



DM503

Forelæsning 4

Indhold

- Træer
 - Hurtig gennemgang af binære søgetræer fra sidst
 - Indsætning og Sletning i binære søgetræer
- 1. delprojekt

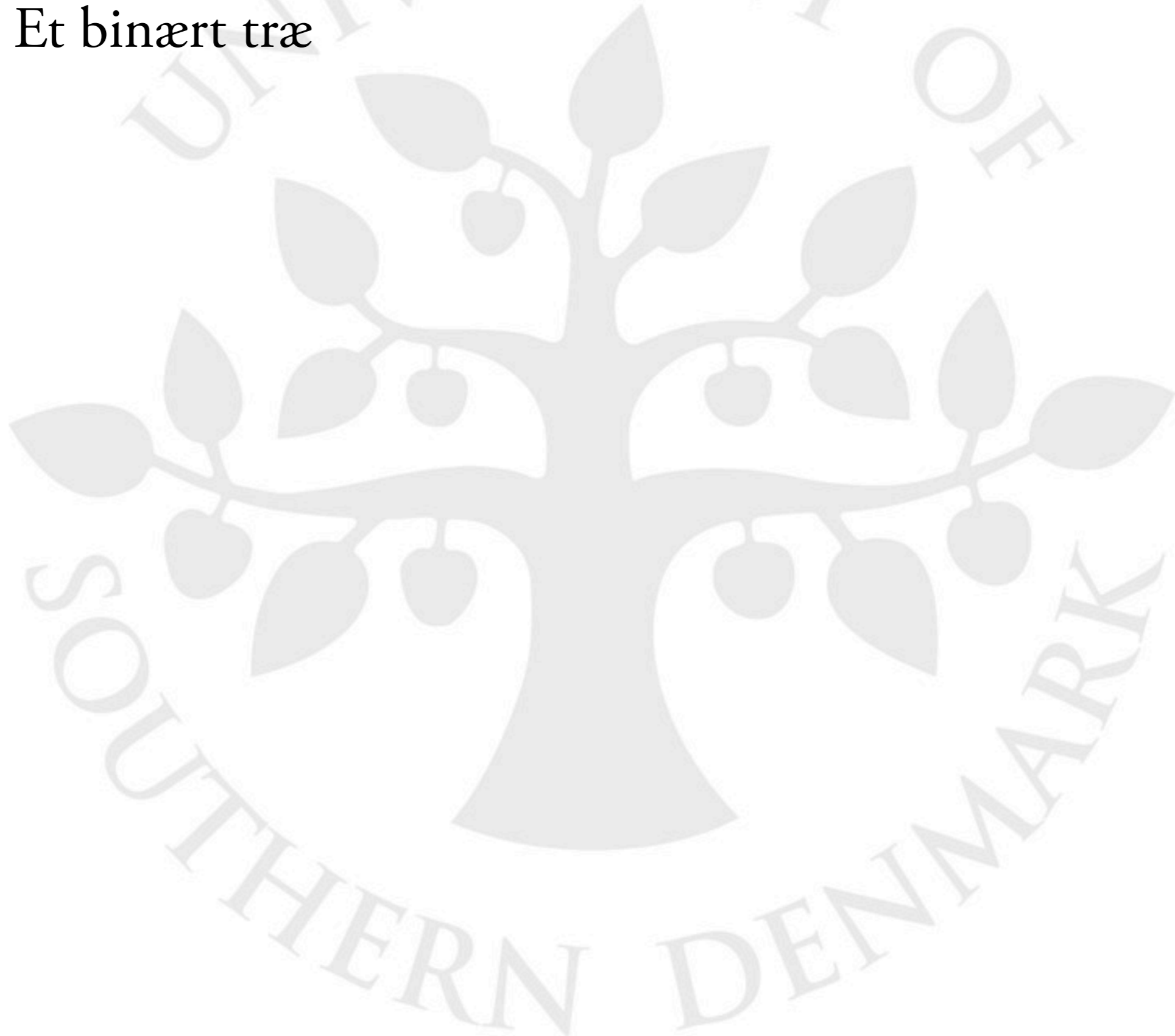


Binære søgetræer



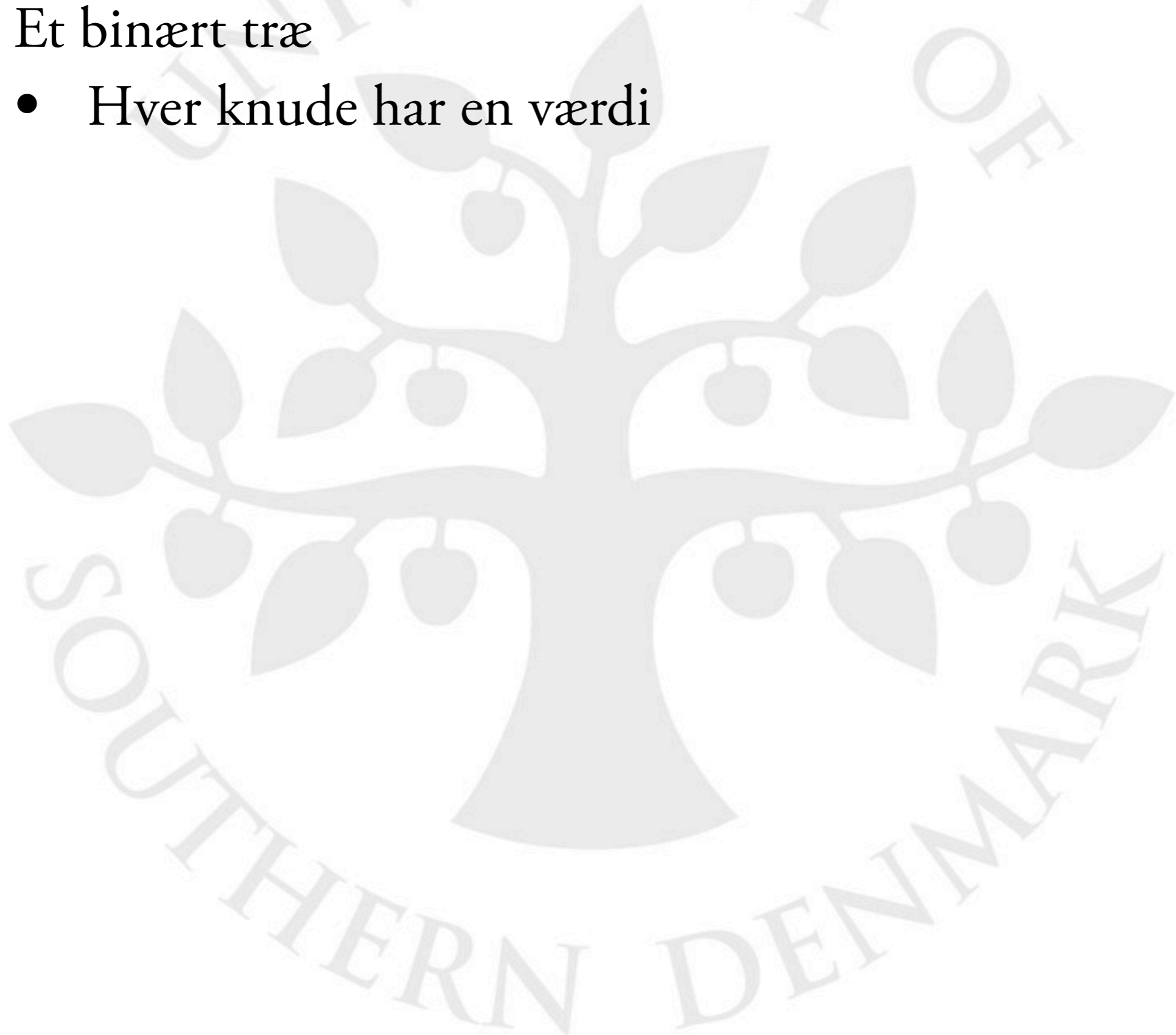
Binære søgetræer

- Et binært træ



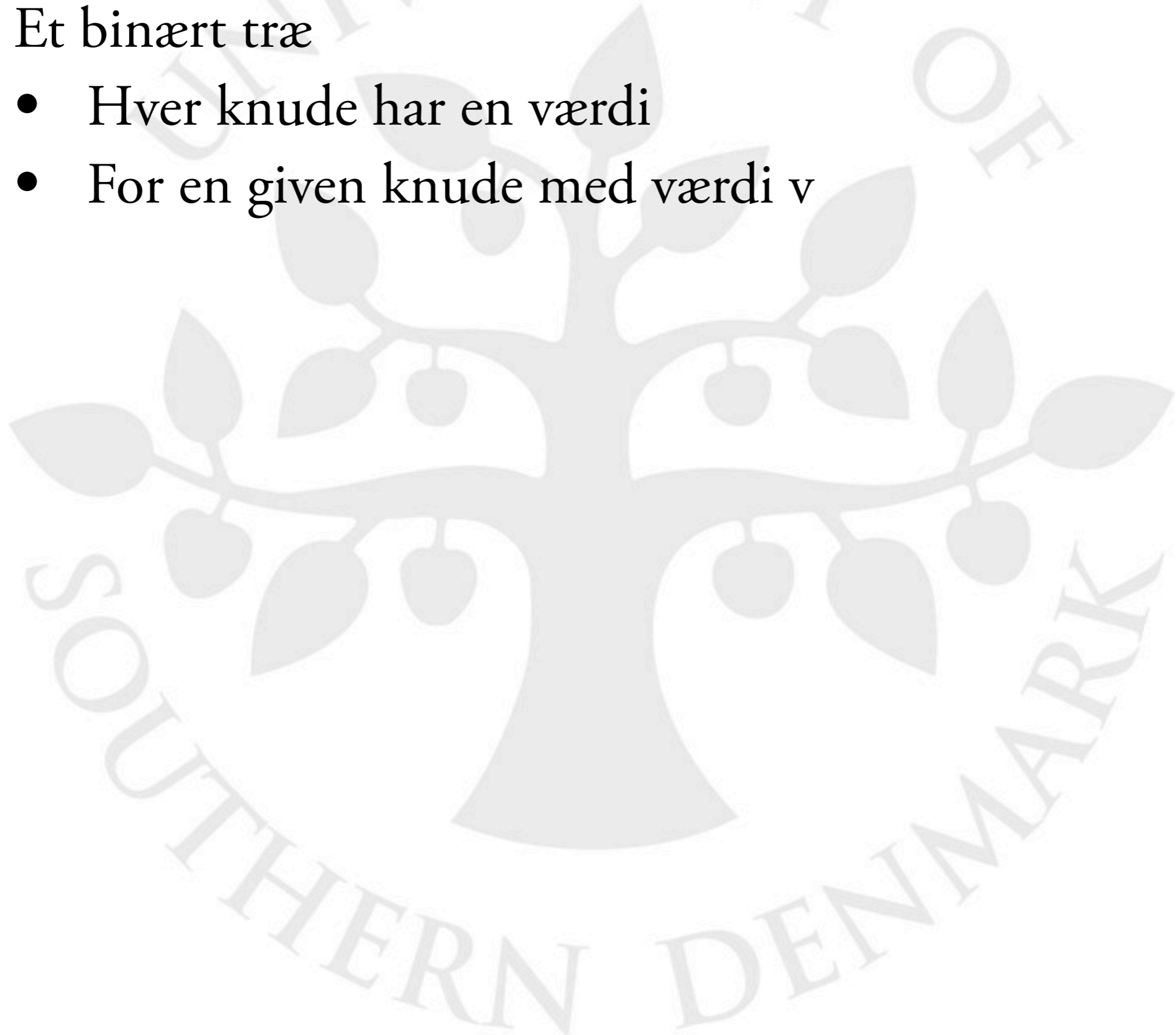
Binære søgetræer

- Et binært træ
 - Hver knude har en værdi



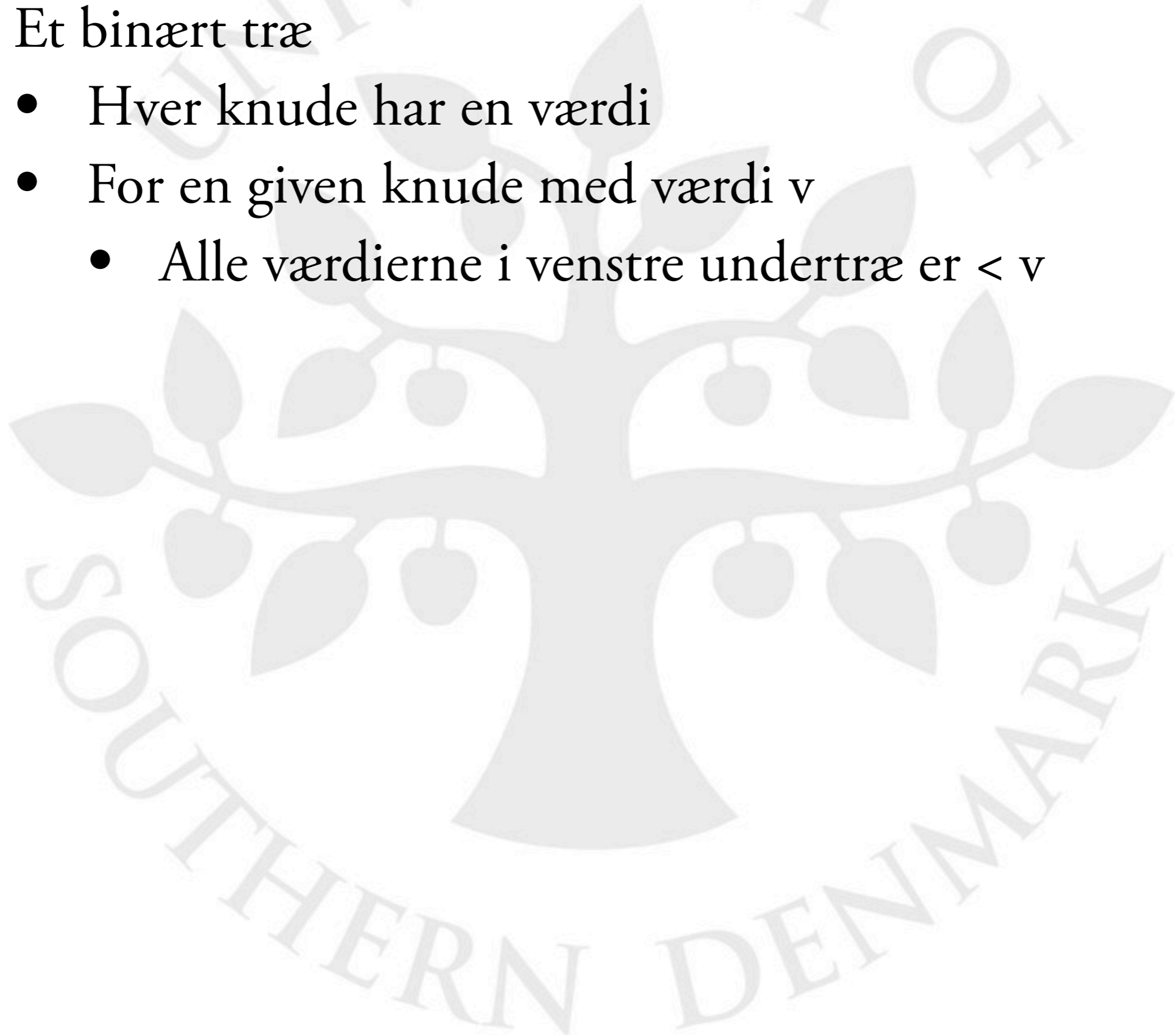
Binære søgetræer

- Et binært træ
 - Hver knude har en værdi
 - For en given knude med værdi v



Binære søgetræer

- Et binært træ
 - Hver knude har en værdi
 - For en given knude med værdi v
 - Alle værdierne i venstre undertræ er $< v$



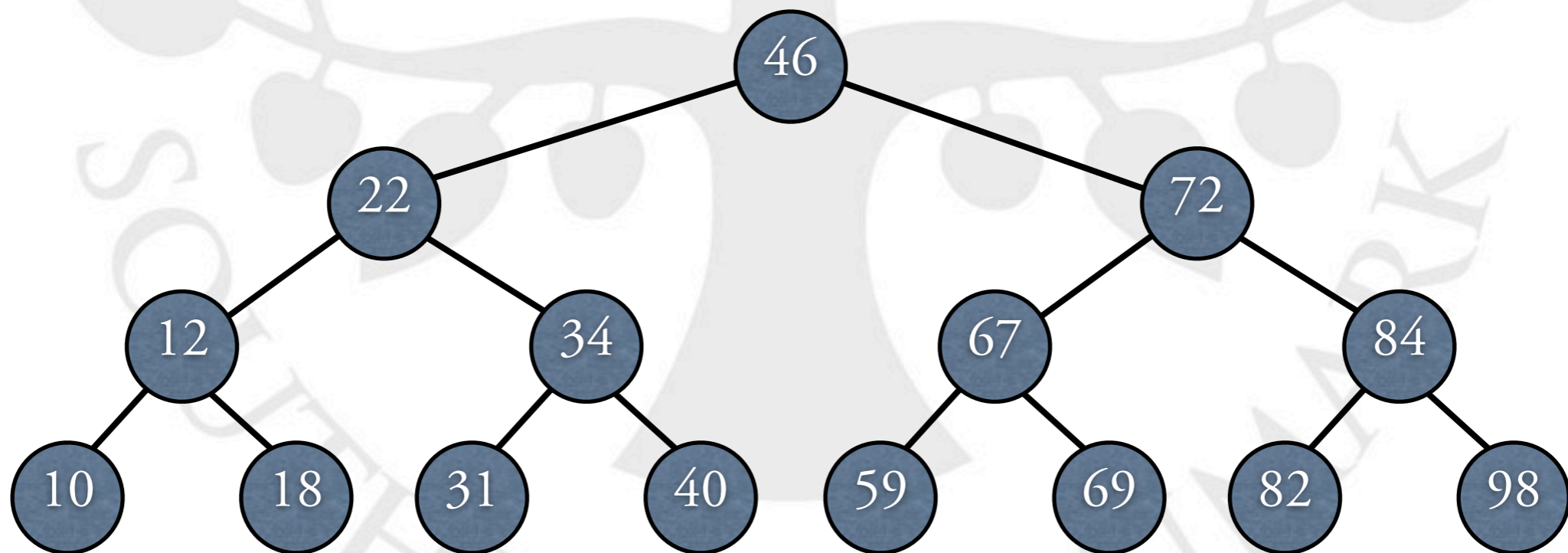
Binære søgetræer

- Et binært træ
 - Hver knude har en værdi
 - For en given knude med værdi v
 - Alle værdierne i venstre undertræ er $< v$
 - Alle værdierne i højre undertræ er $> v$

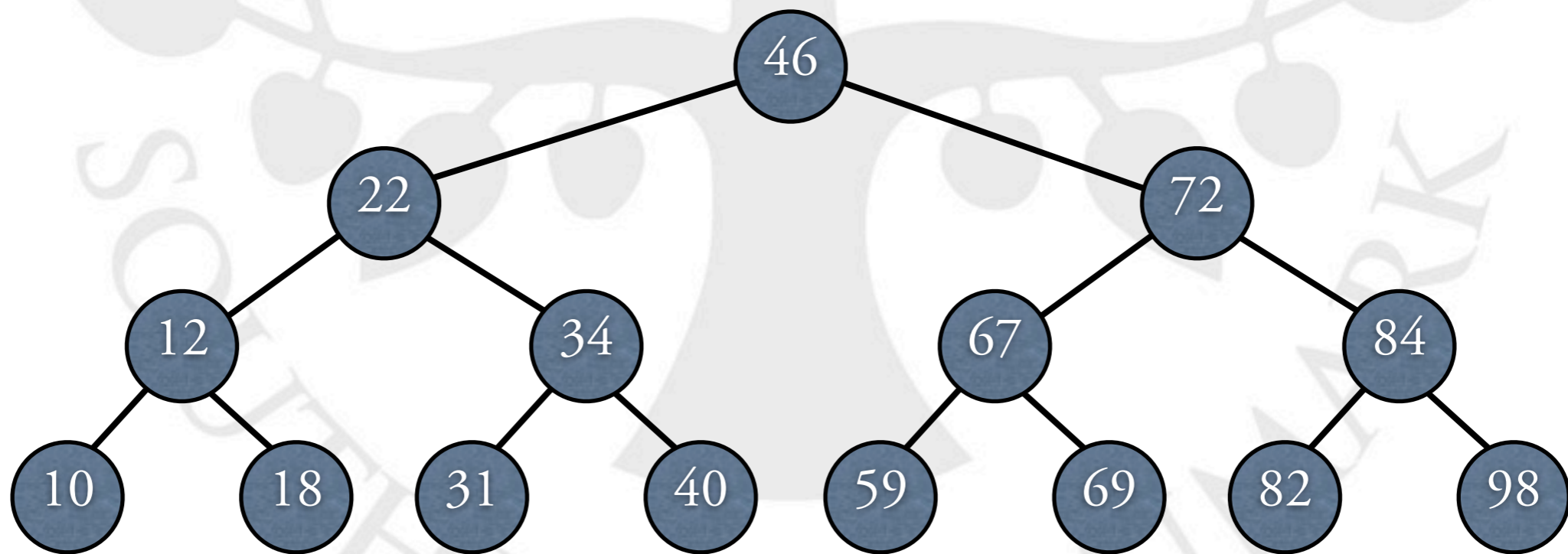


Binære søgetræer

- Et binært træ
 - Hver knude har en værdi
 - For en given knude med værdi v
 - Alle værdierne i venstre undertræ er $< v$
 - Alle værdierne i højre undertræ er $> v$

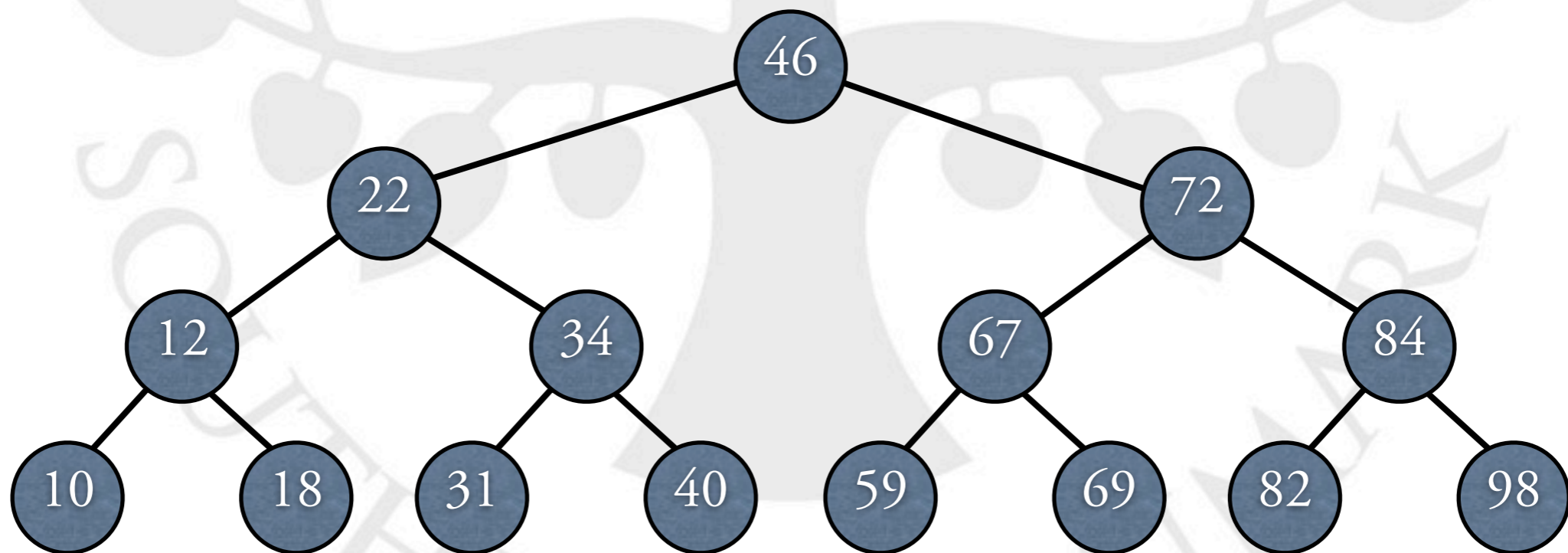


Binære søgetræer



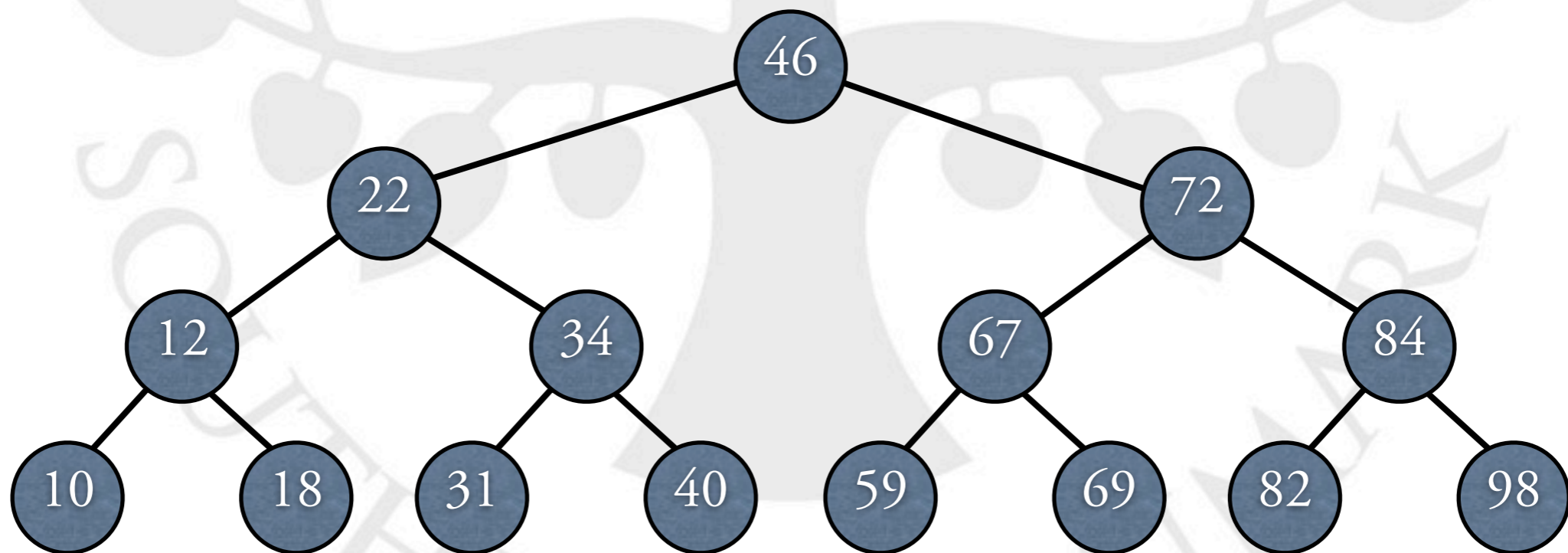
Binære søgetræer

- Søgning



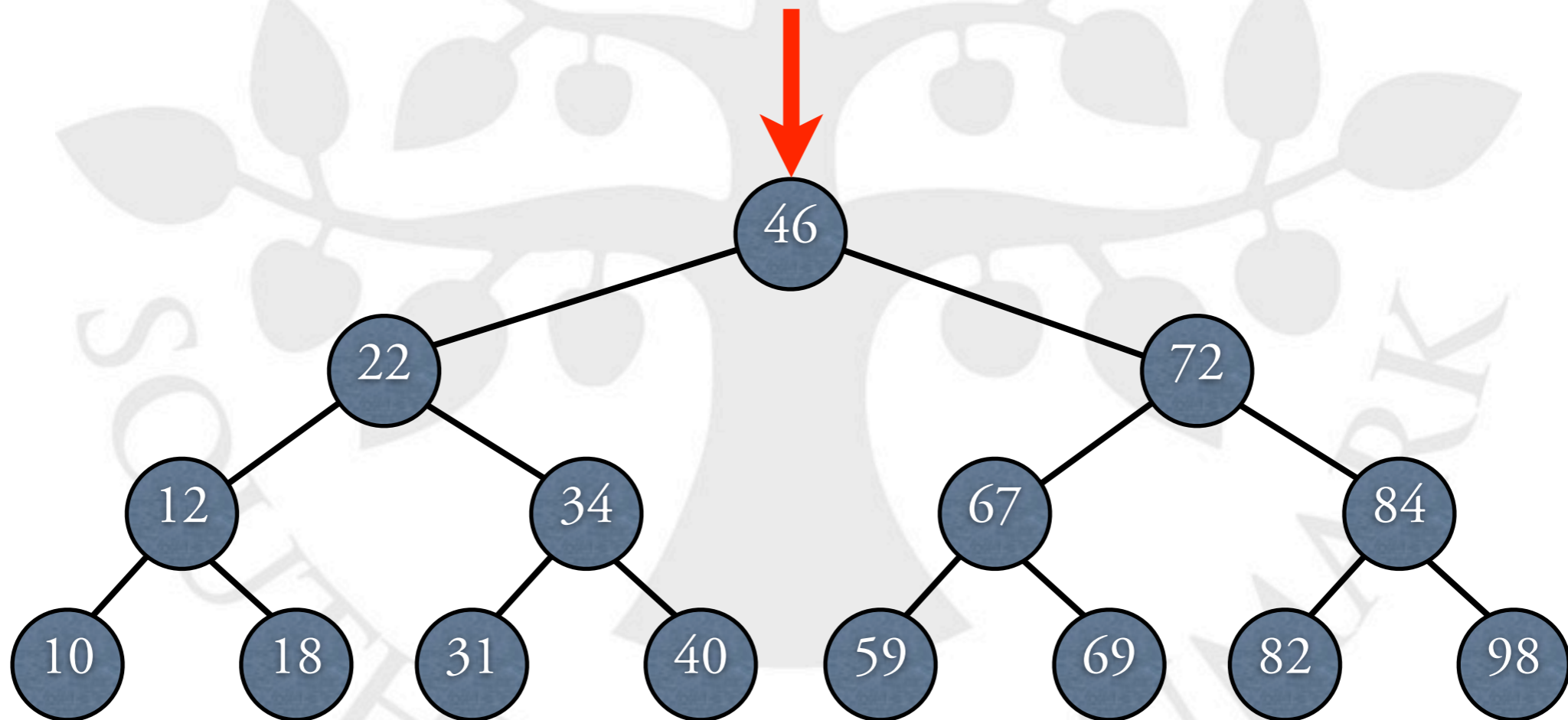
Binære søgetræer

- Søgning
 - Search(82)



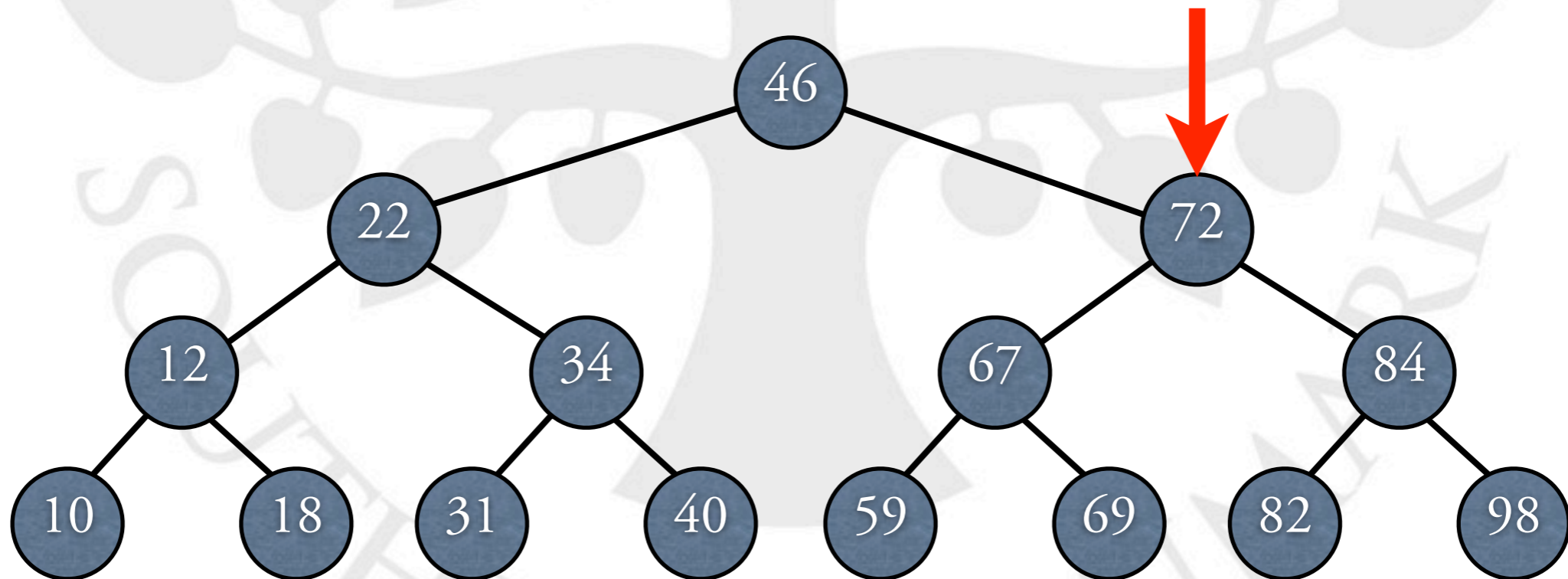
Binære søgetræer

- Søgning
 - Search(82)



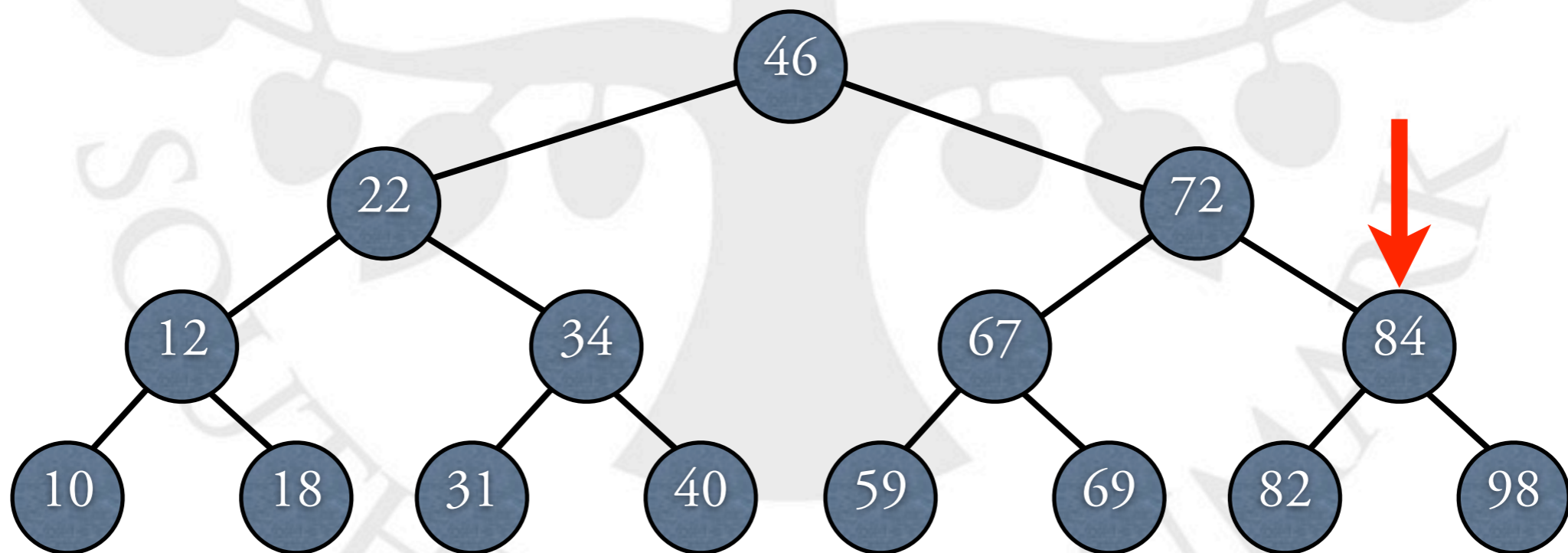
Binære søgetræer

- Søgning
 - Search(82)



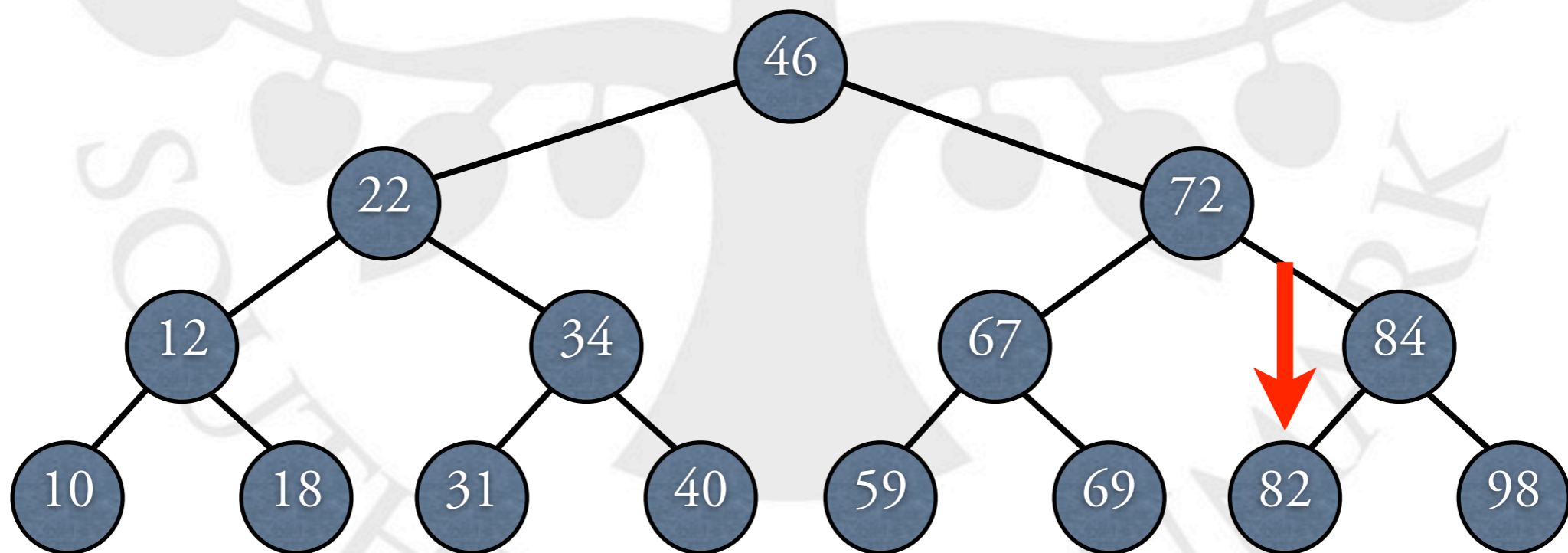
Binære søgetræer

- Søgning
 - Search(82)



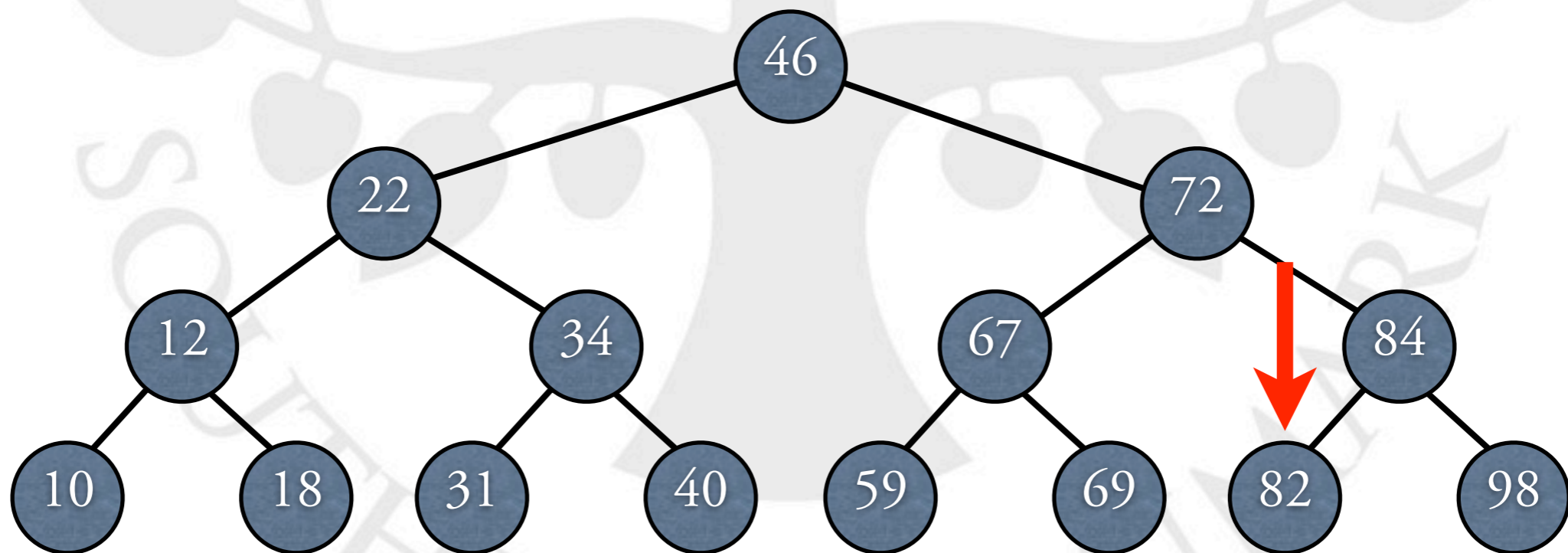
Binære søgetræer

- Søgning
 - Search(82)



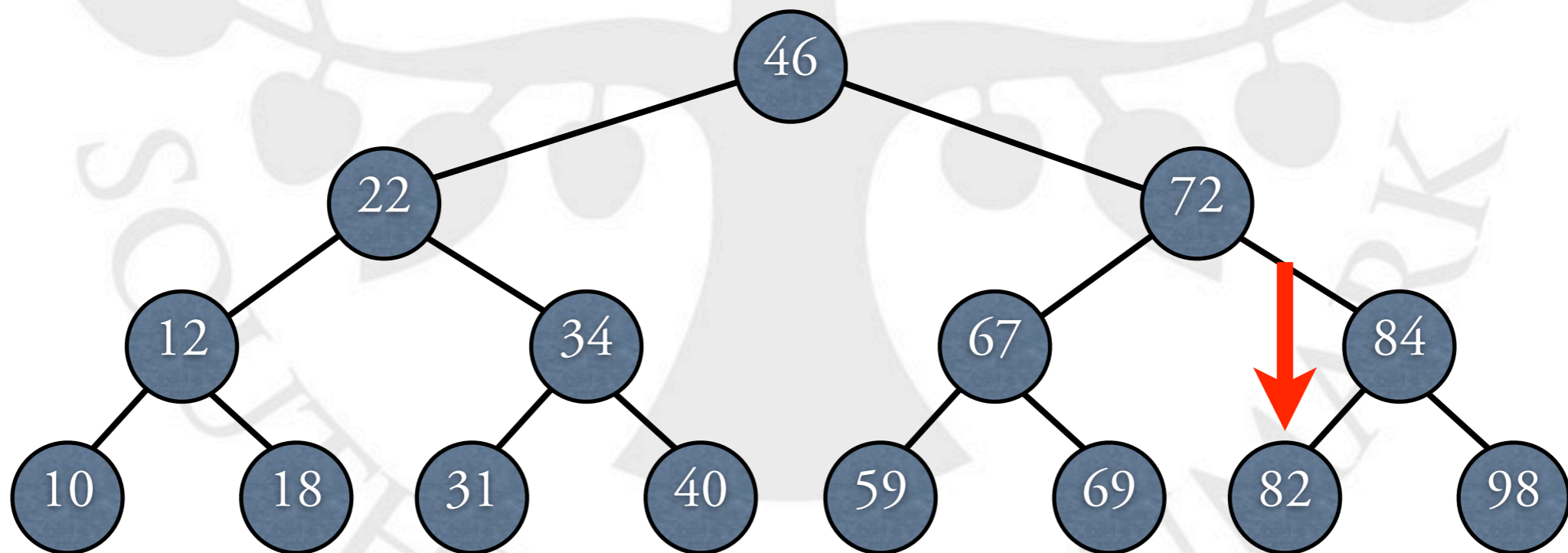
Binære søgetræer

- Søgning
 - Search(82)
 - Lige så let som binær søgning

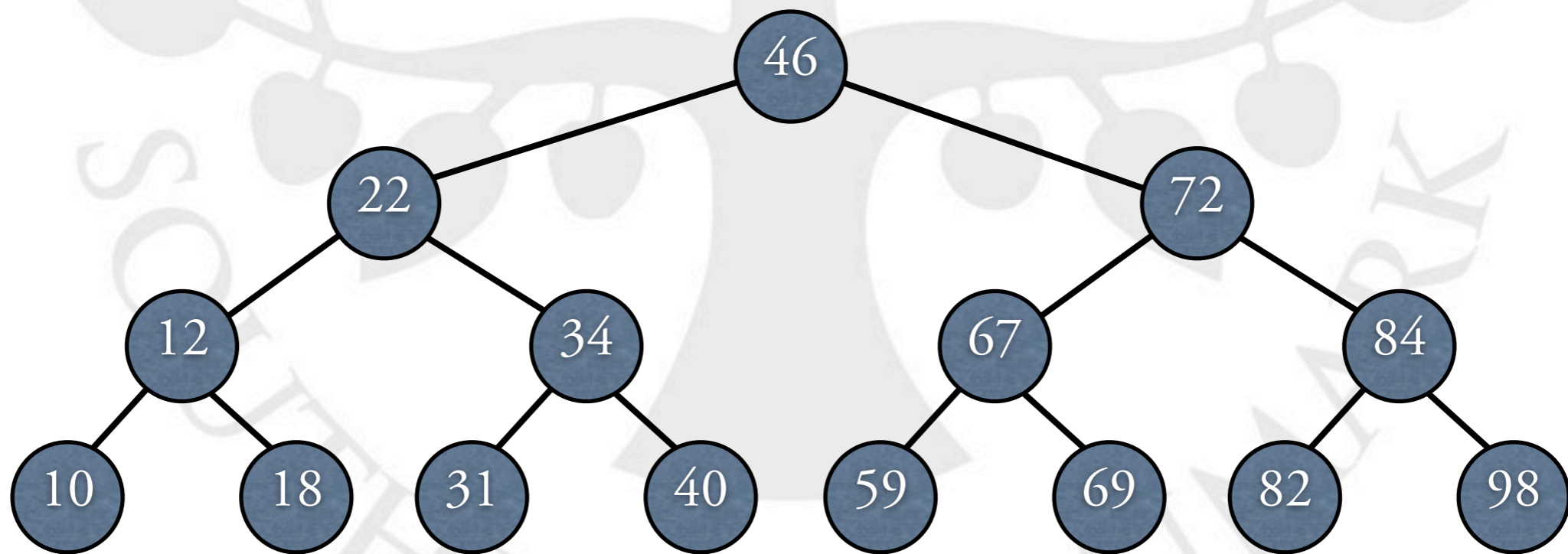


Binære søgetræer

- Søgning
 - Search(82)
 - Lige så let som binær søgning
 - Vi gør faktisk præcis det samme

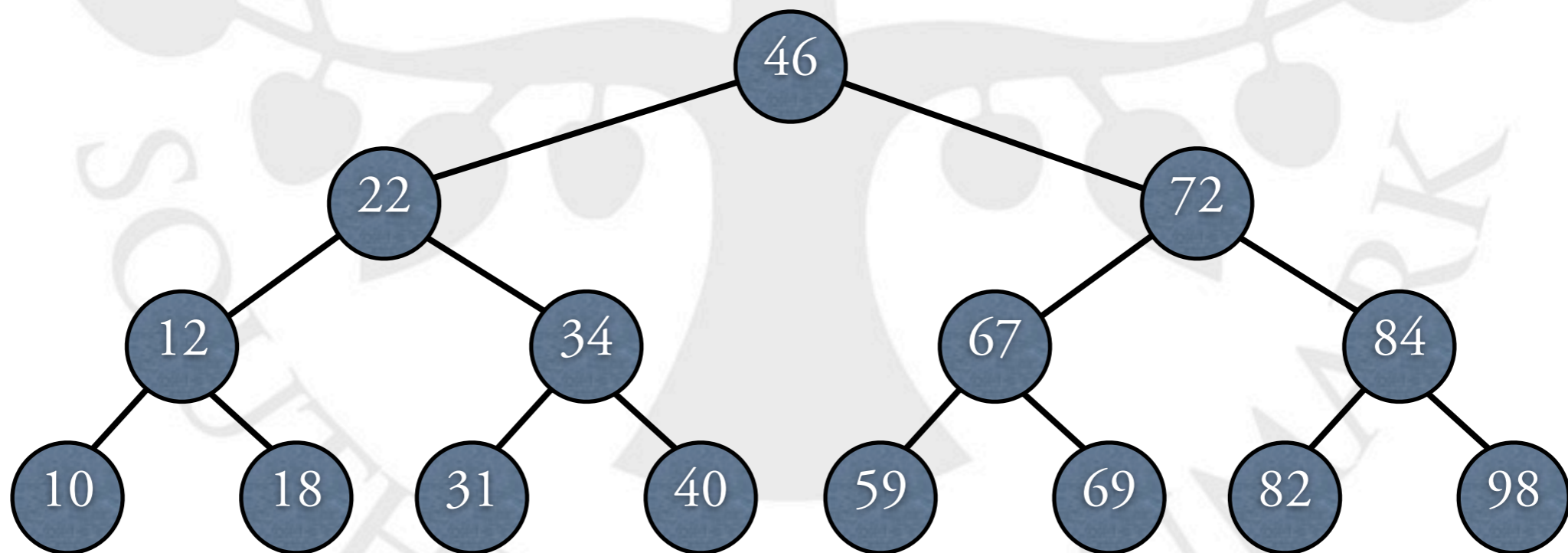


Binære søgetræer



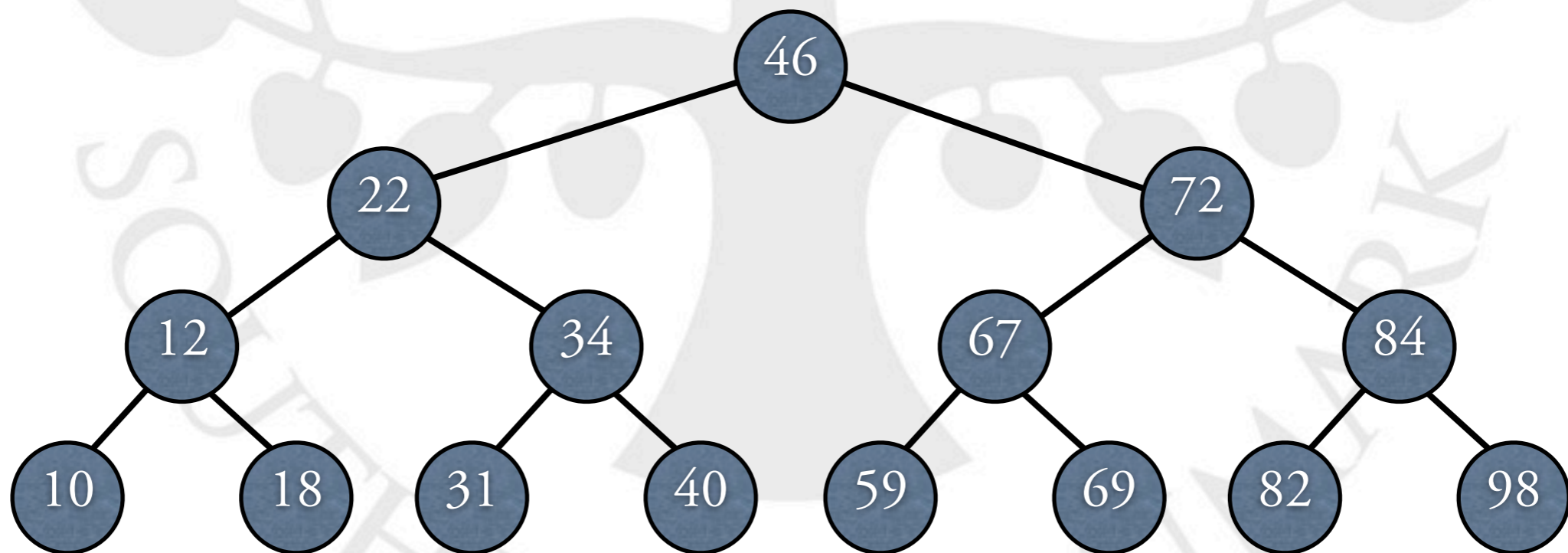
Binære søgetræer

- Hvad nu med indsættelser?



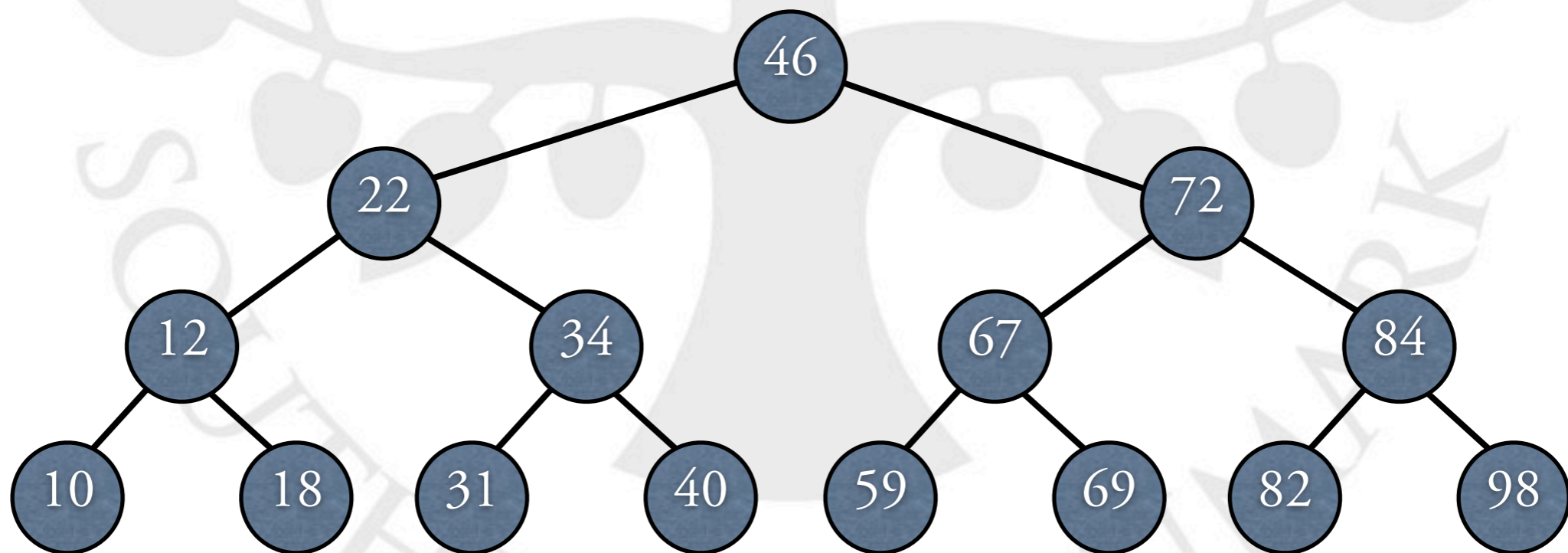
Binære søgetræer

- Hvad nu med indsættelser?
 - Insert(42)



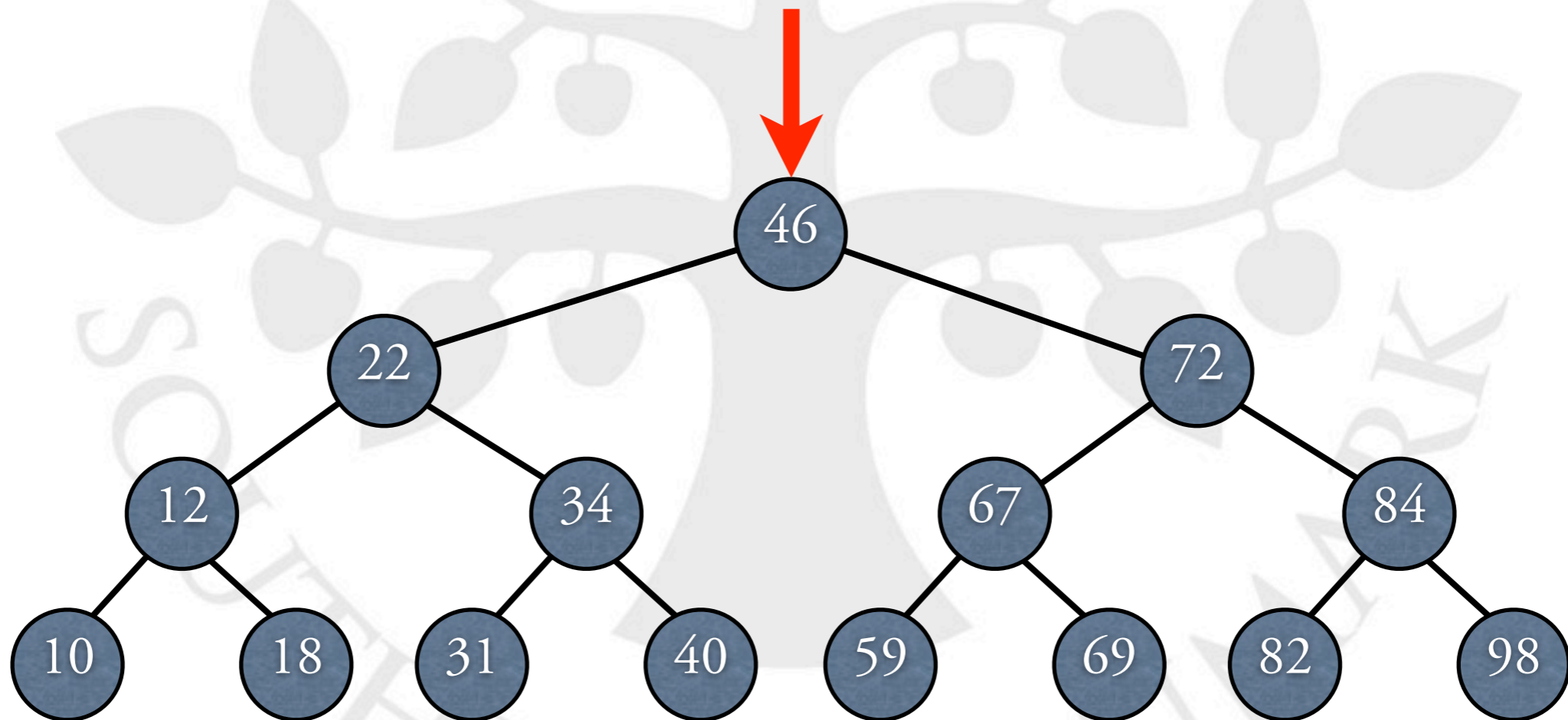
Binære søgetræer

- Hvad nu med indsættelser?
 - Insert(42)
 - Findes 42 allerede? (Search(42))



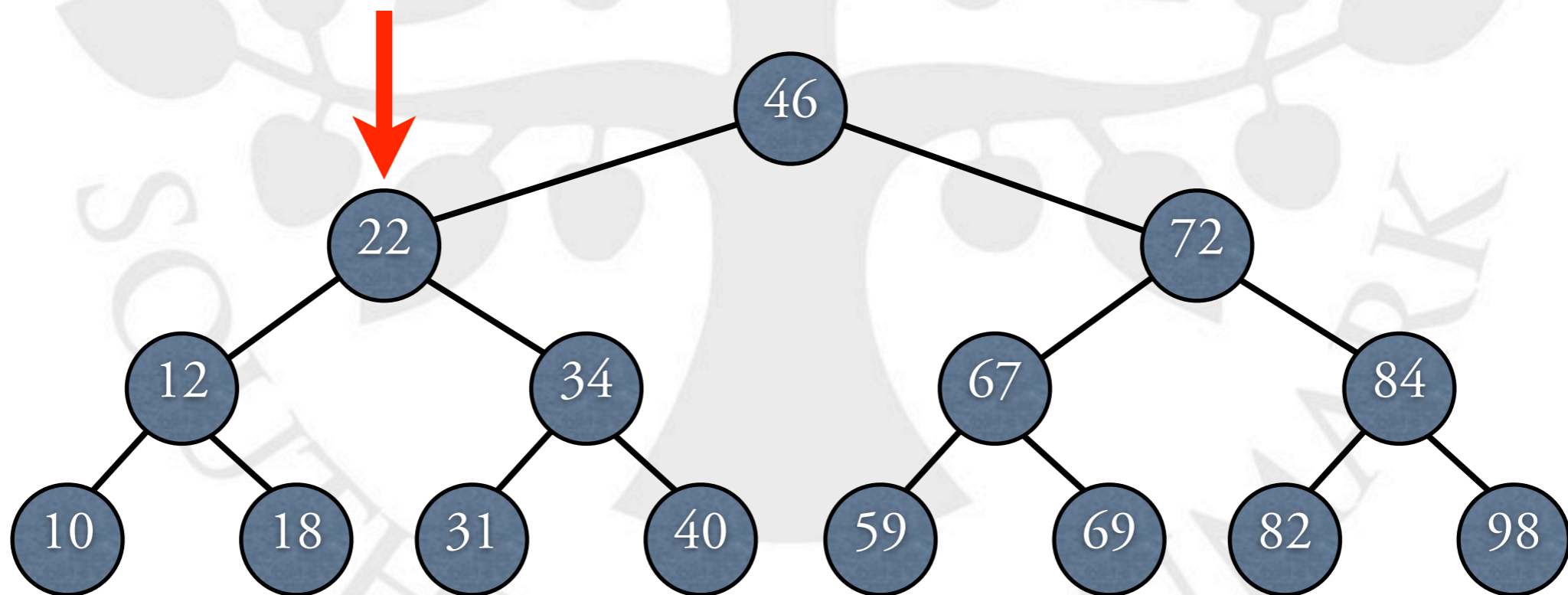
Binære søgetræer

- Hvad nu med indsættelser?
 - Insert(42)
 - Findes 42 allerede? (Search(42))



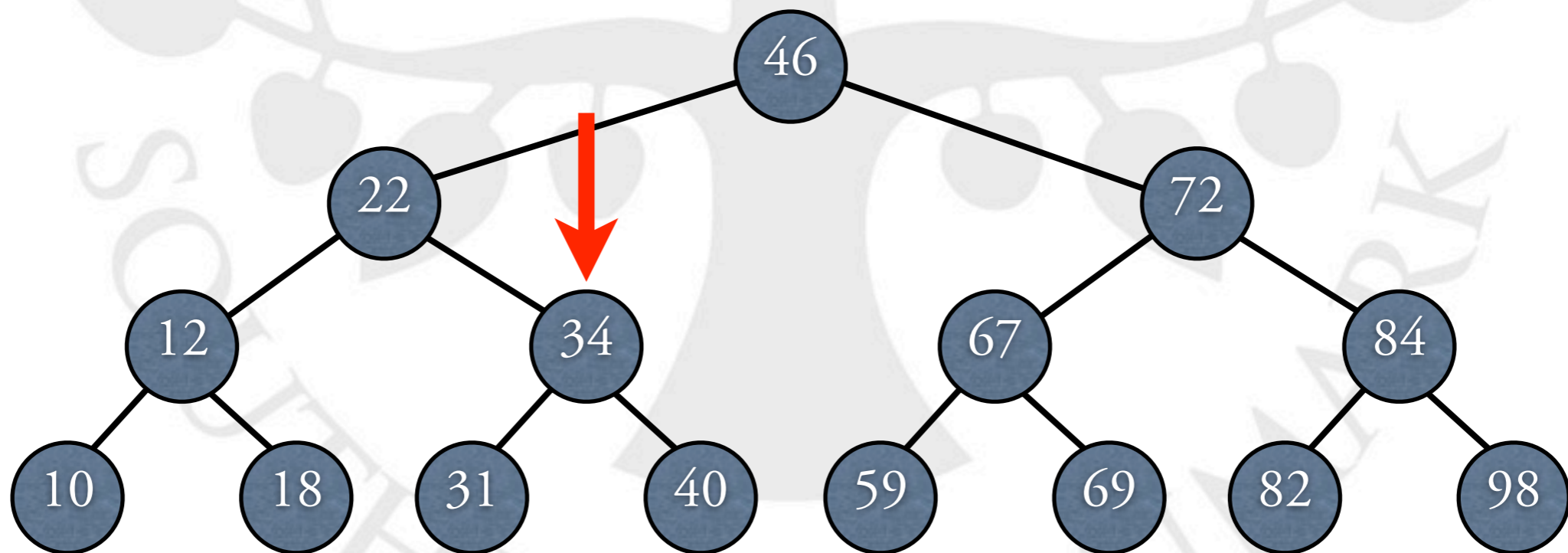
Binære søgetræer

- Hvad nu med indsættelser?
 - Insert(42)
 - Findes 42 allerede? (Search(42))



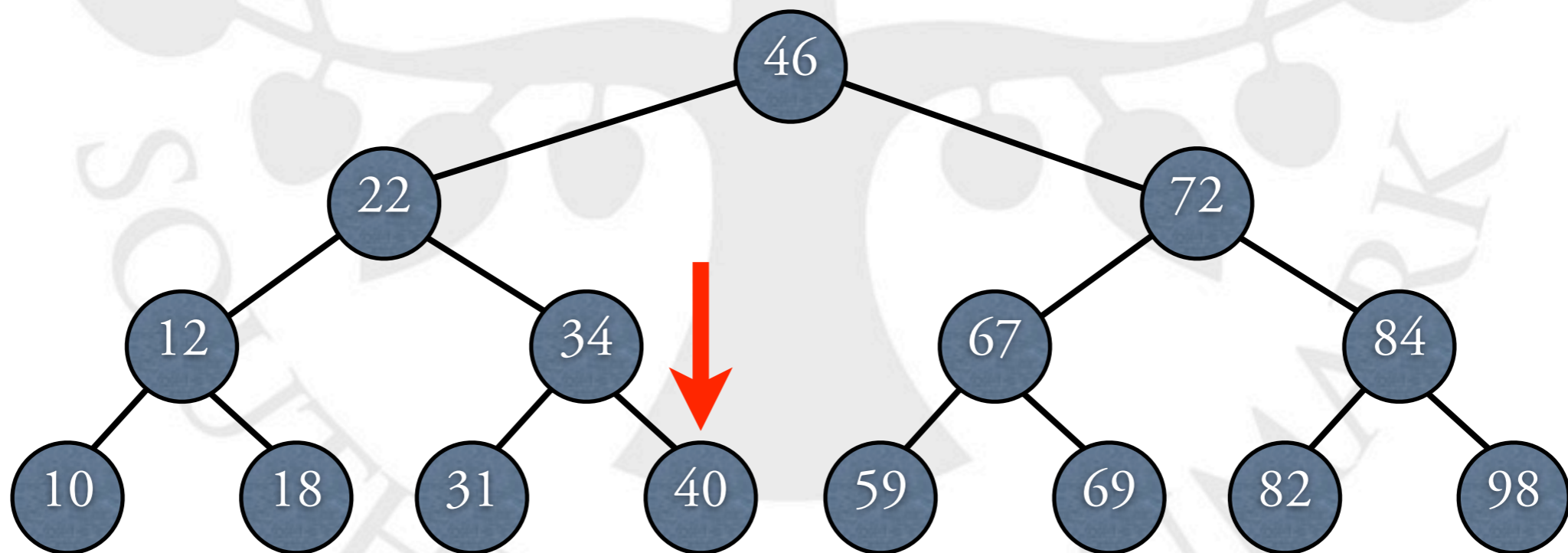
Binære søgetræer

- Hvad nu med indsættelser?
 - Insert(42)
 - Findes 42 allerede? (Search(42))



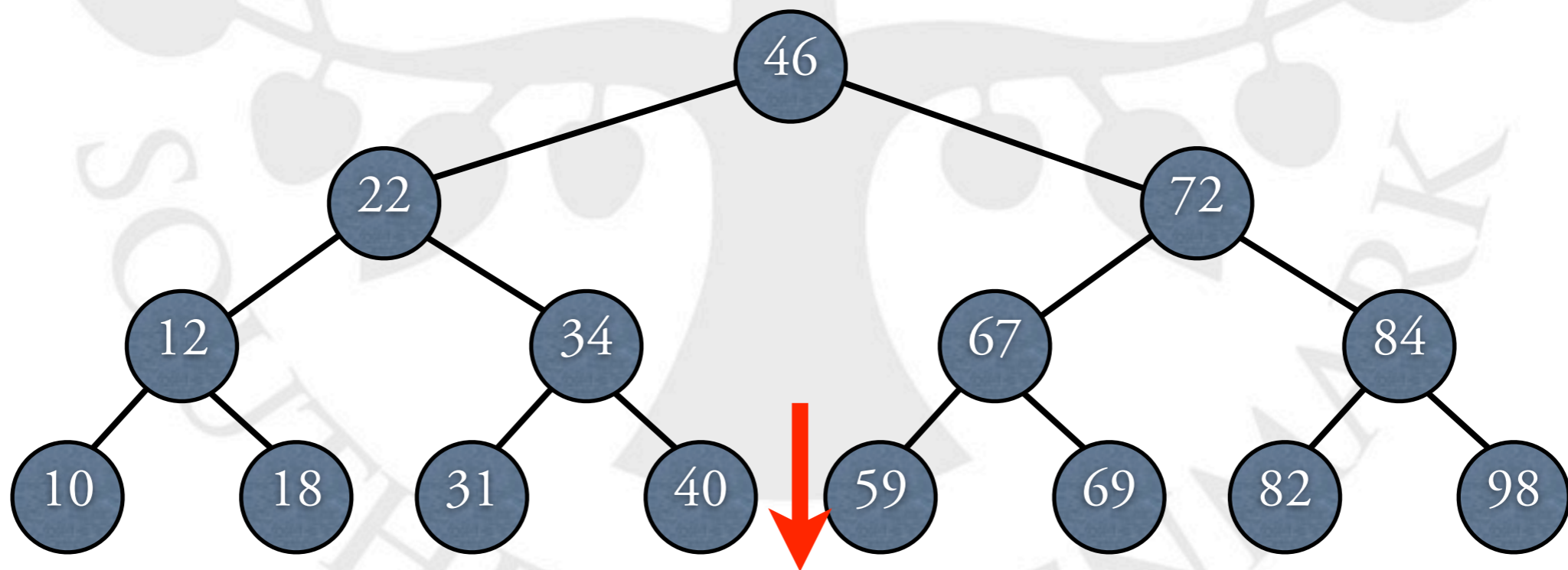
Binære søgetræer

- Hvad nu med indsættelser?
 - Insert(42)
 - Findes 42 allerede? (Search(42))



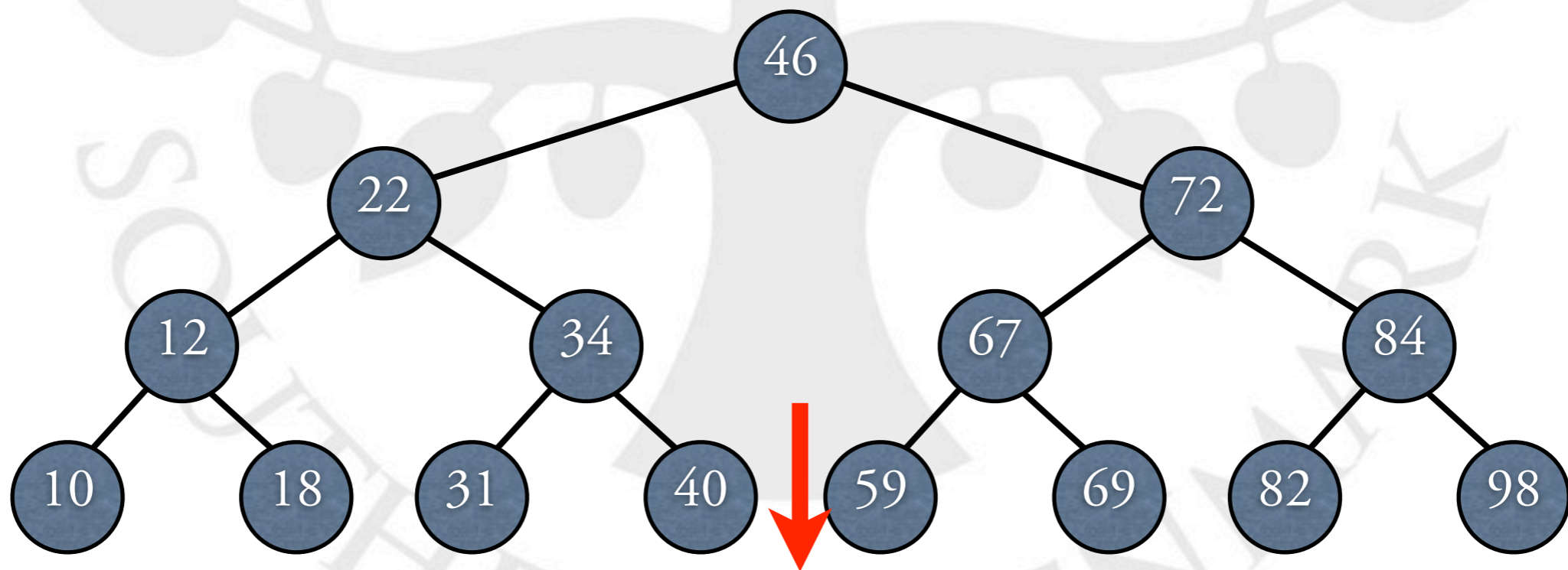
Binære søgetræer

- Hvad nu med indsættelser?
 - Insert(42)
 - Findes 42 allerede? (Search(42))



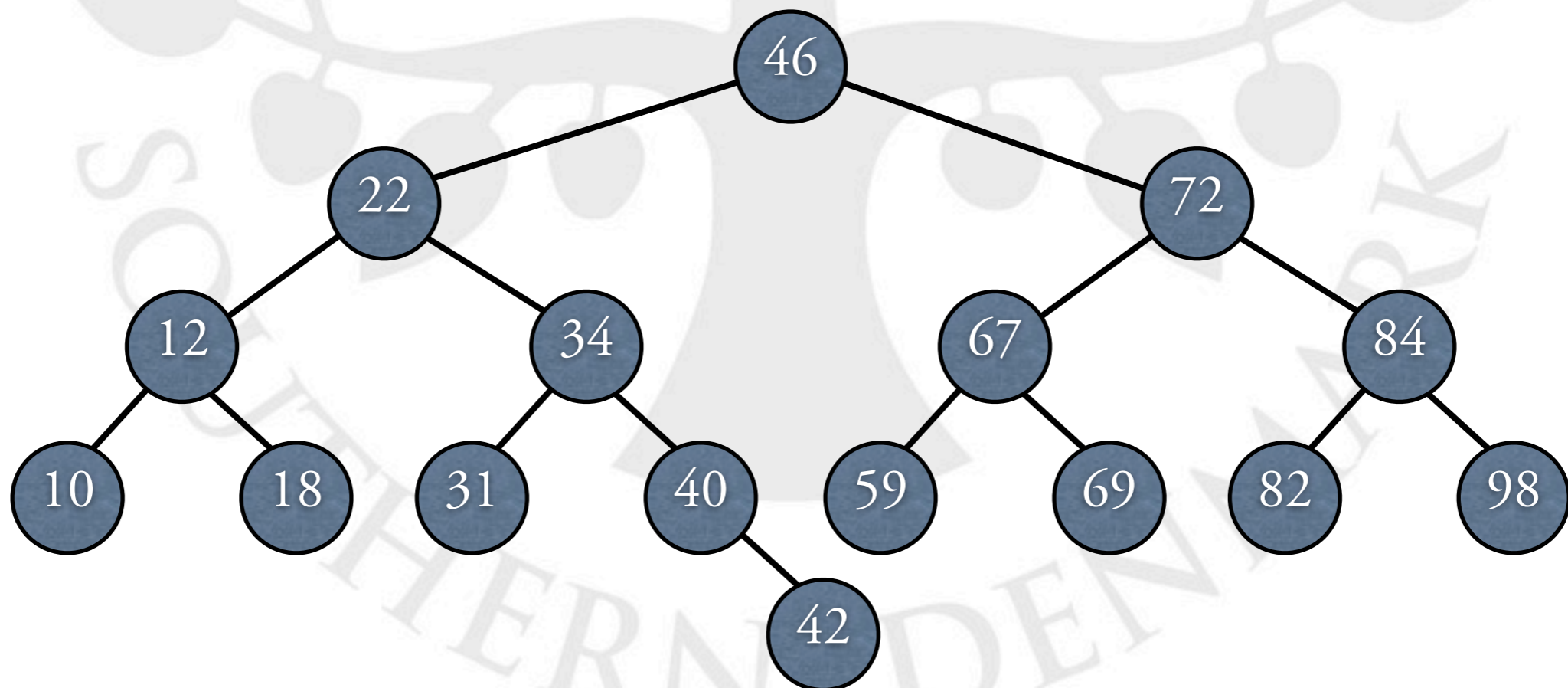
Binære søgetræer

- Hvad nu med indsættelser?
 - Insert(42)
 - Findes 42 allerede? (Search(42))
 - Nu ved vi hvor 42 bør sidde i træet

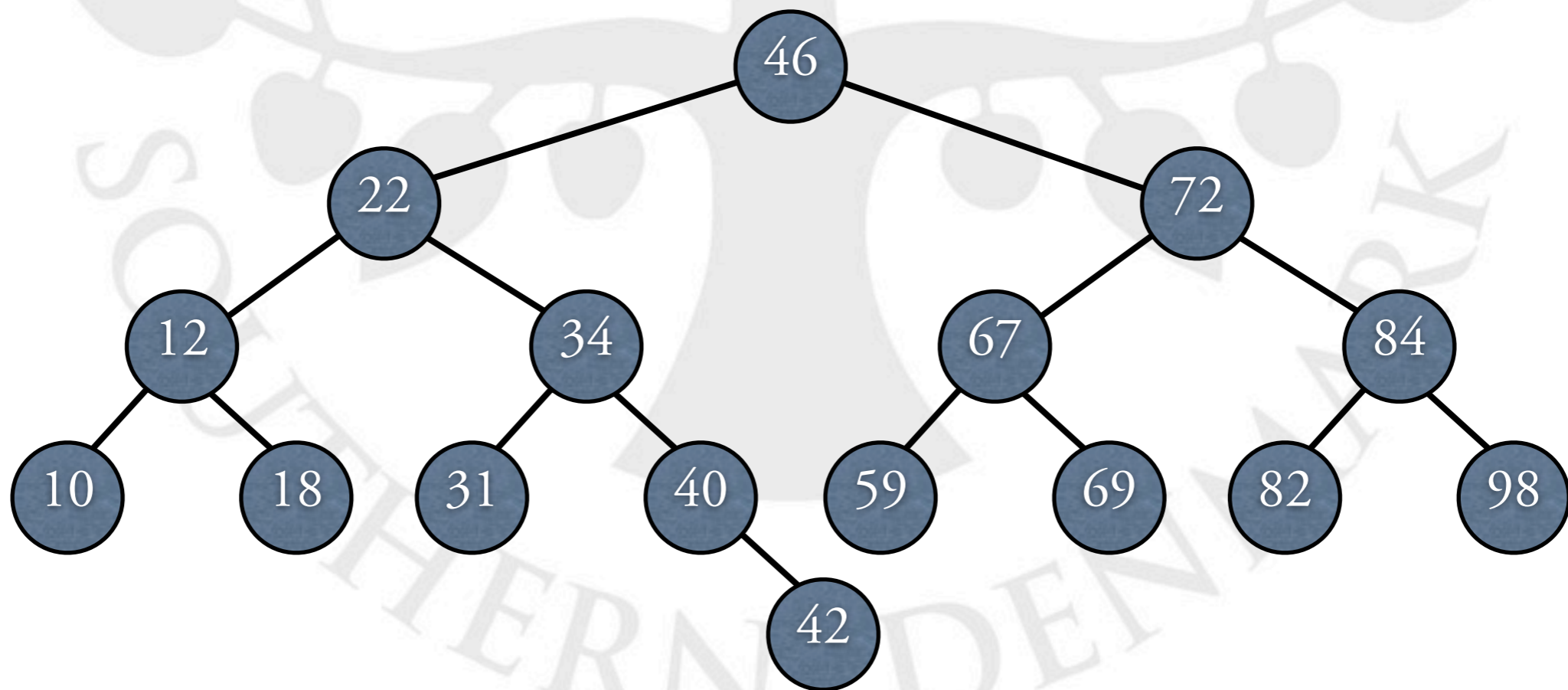


Binære søgetræer

- Hvad nu med indsættelser?
 - Insert(42)
 - Findes 42 allerede? (Search(42))
 - Nu ved vi hvor 42 bør sidde i træet

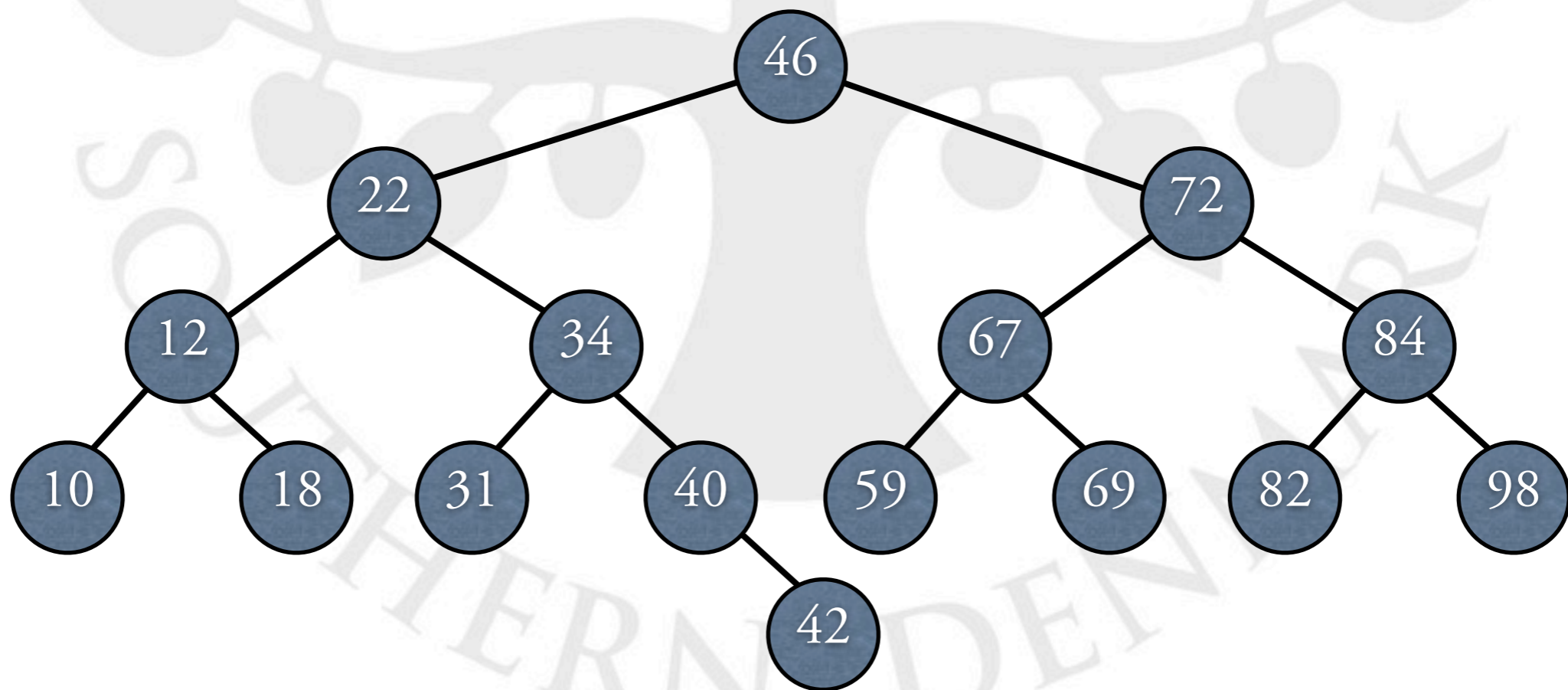


Binære søgetræer



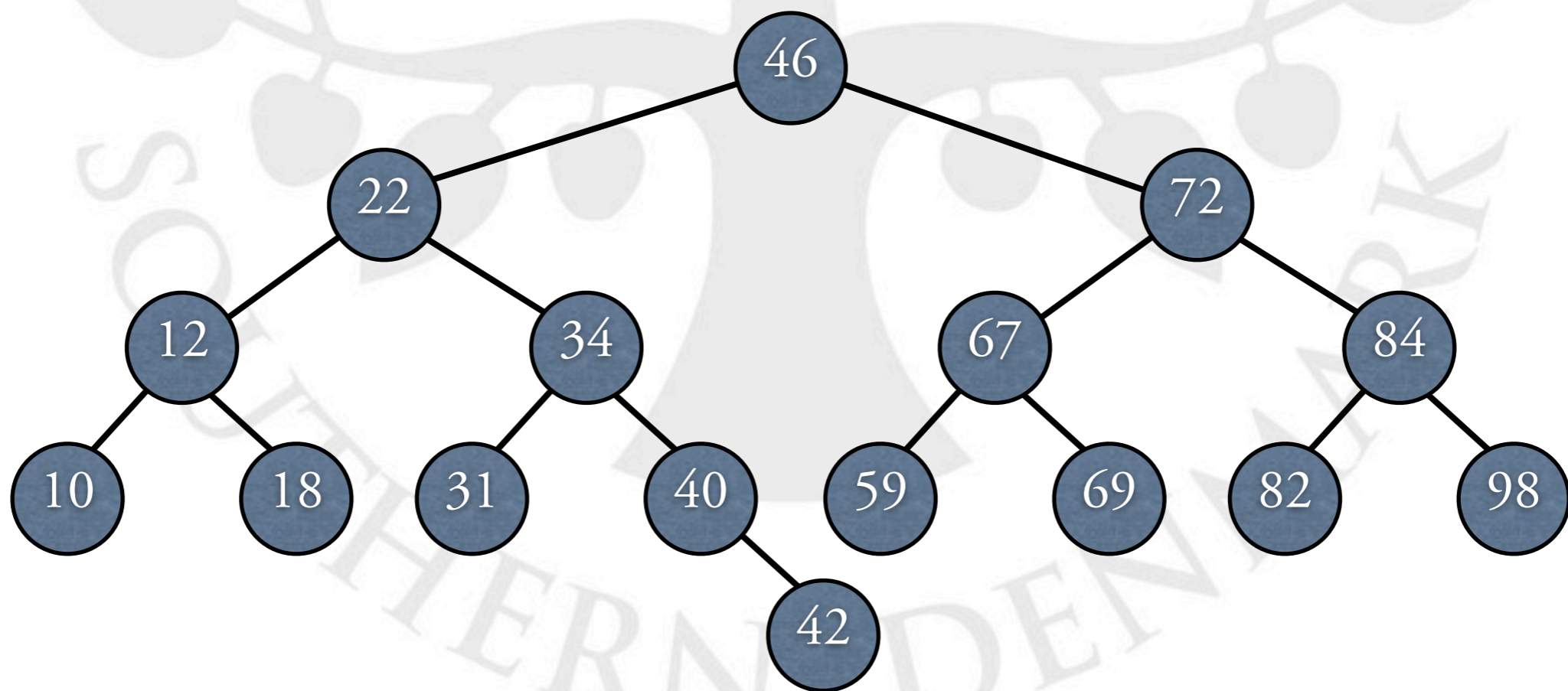
Binære søgetræer

- Hvad med sletninger?



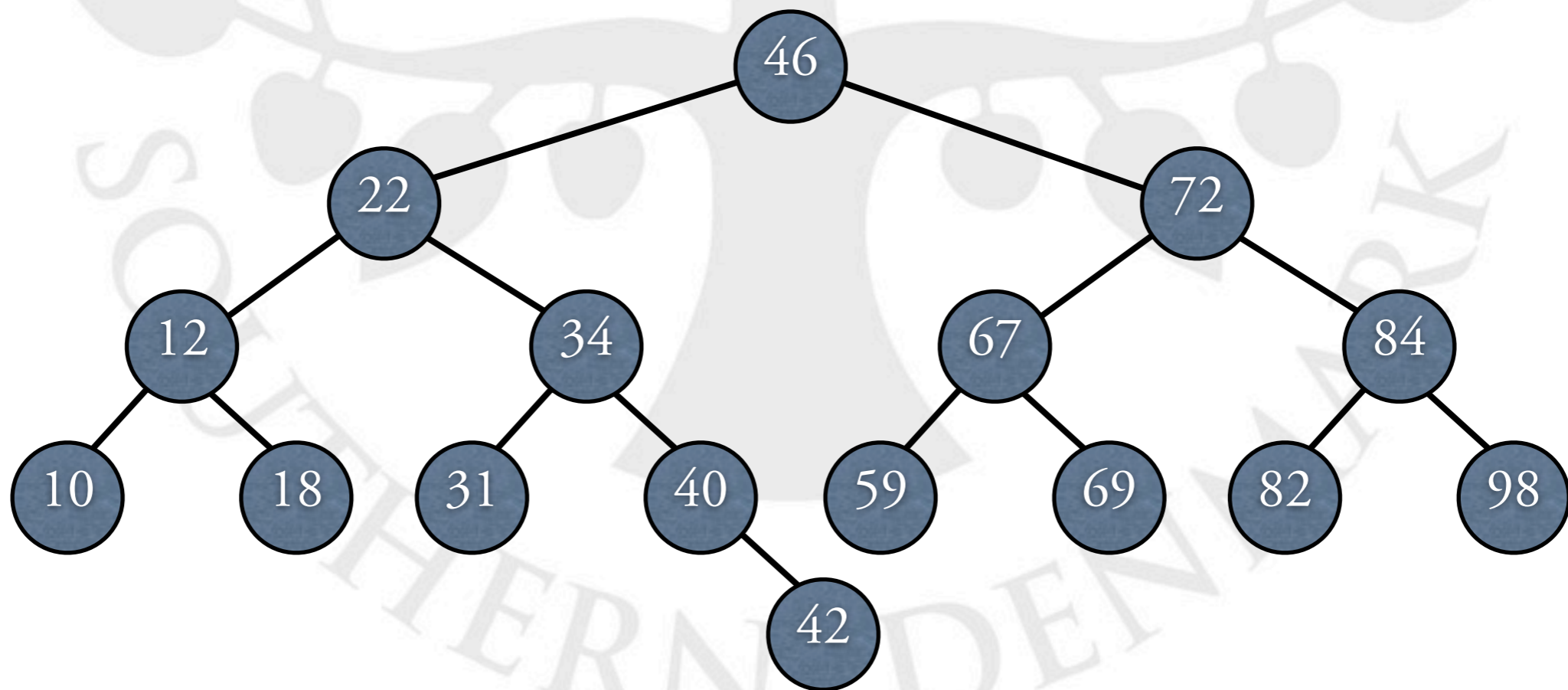
Binære søgetræer

- Hvad med sletninger?
 - Hvorfor er sletninger svære end indsættelser?

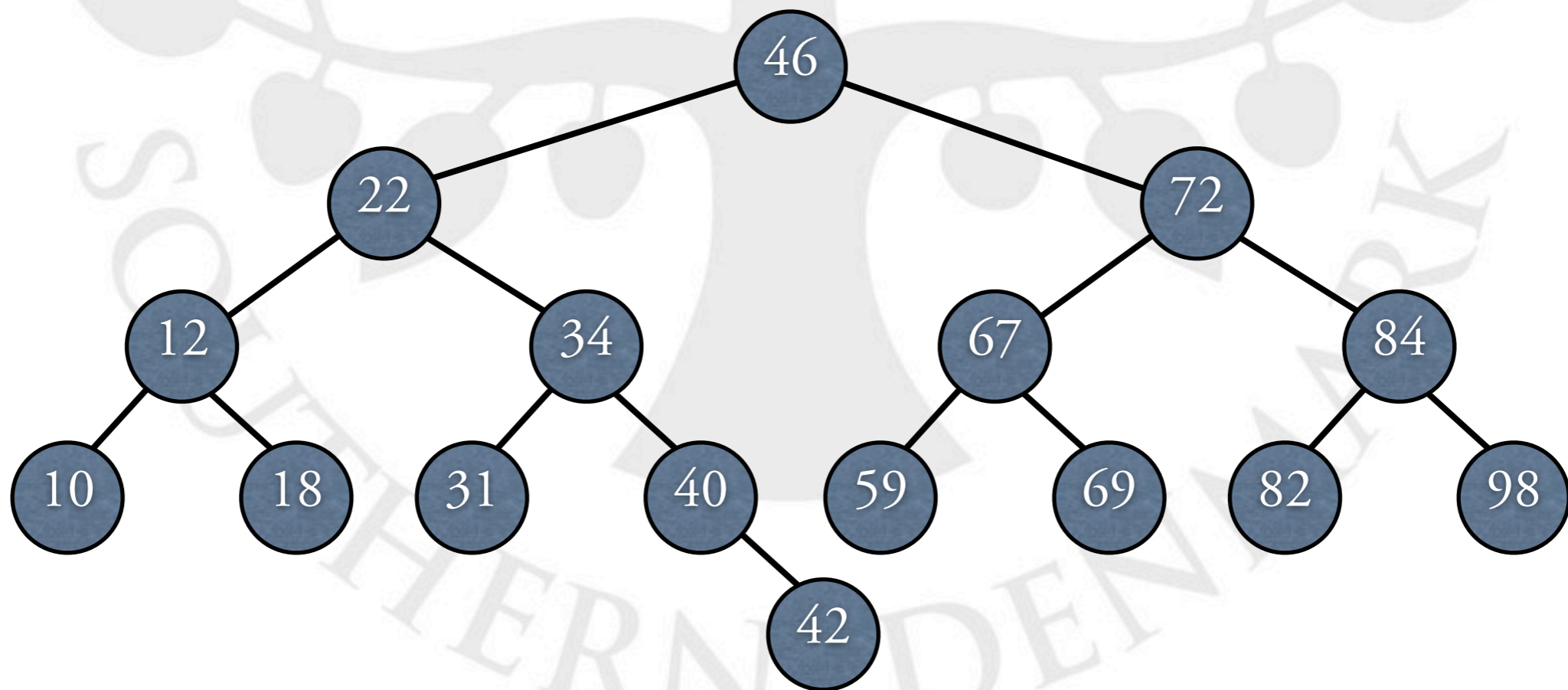


Binære søgetræer

- Hvad med sletninger?
 - Hvorfor er sletninger svære end indsættelser?
 - Skal bruge `prev(e)` og `next(e)`

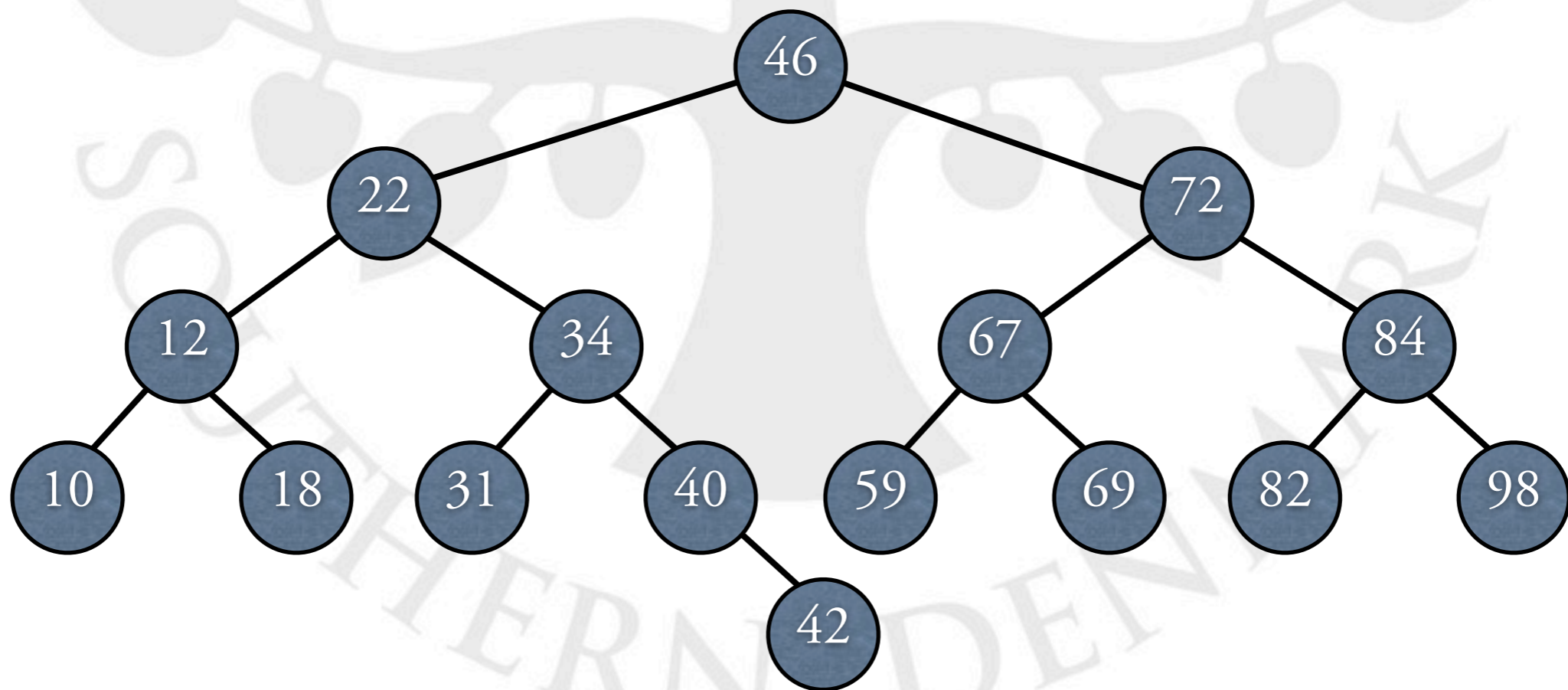


Binære søgetræer



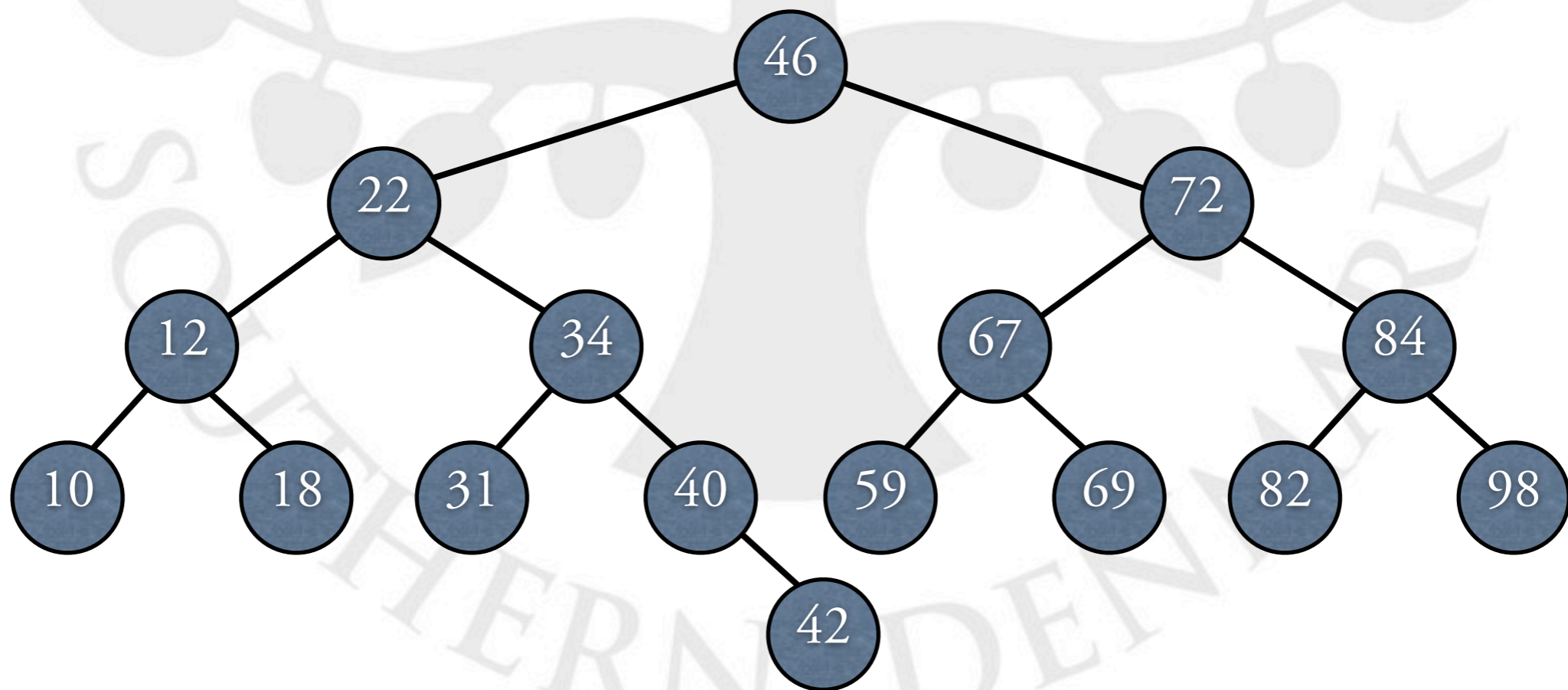
Binære søgetræer

- $prev(e)$ og $next(e)$



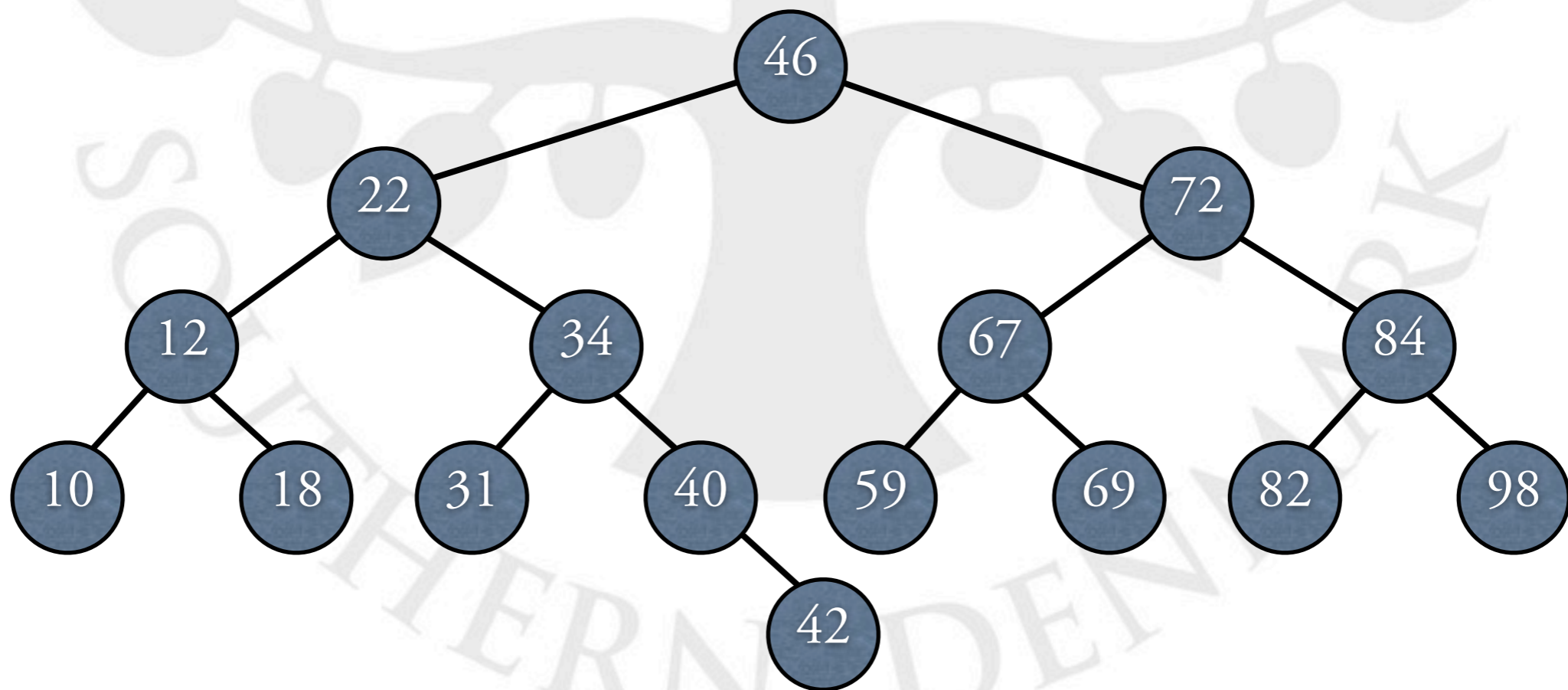
Binære søgetræer

- $\text{prev}(e)$ og $\text{next}(e)$
 - Let i Array og dobbelt-hægtet liste



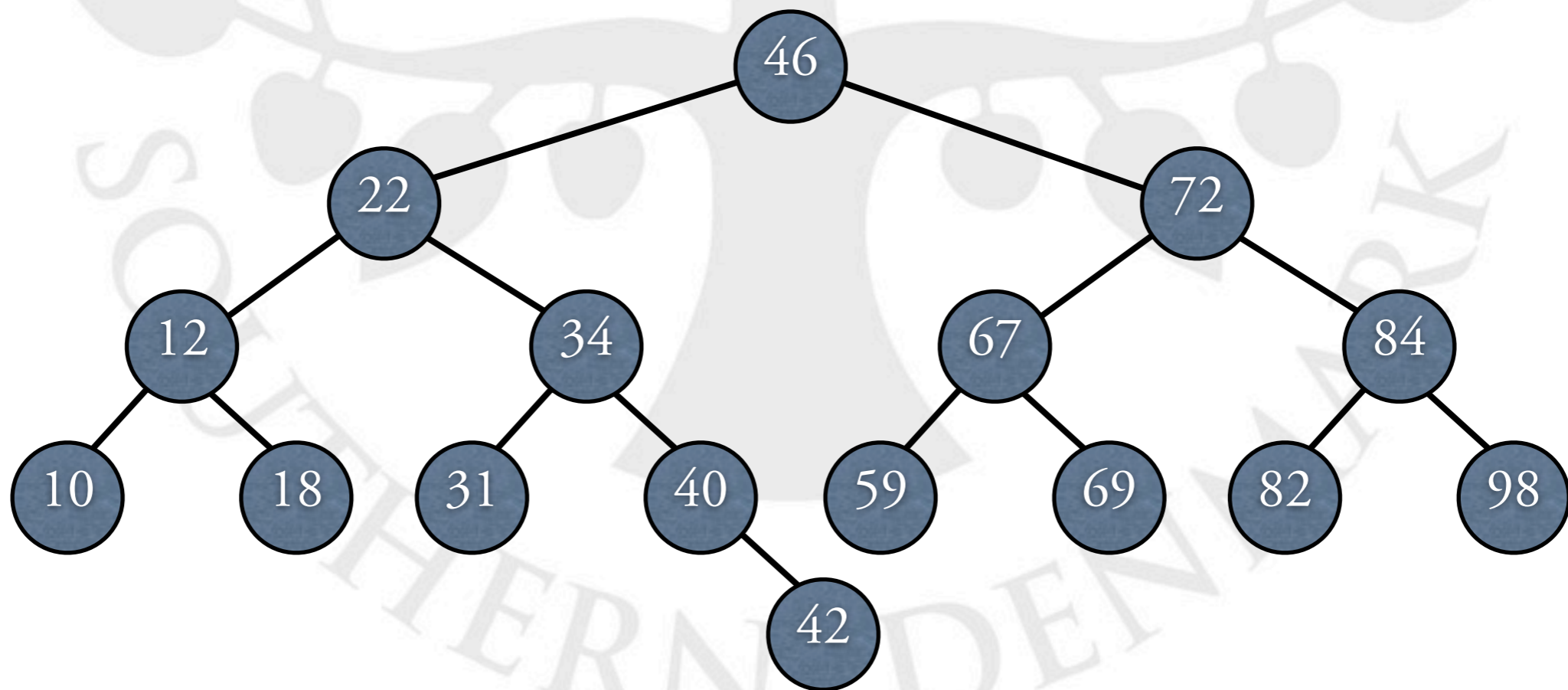
Binære søgetræer

- $\text{prev}(e)$ og $\text{next}(e)$
 - Let i Array og dobbelt-hægtet liste
- $\text{next}(46)$



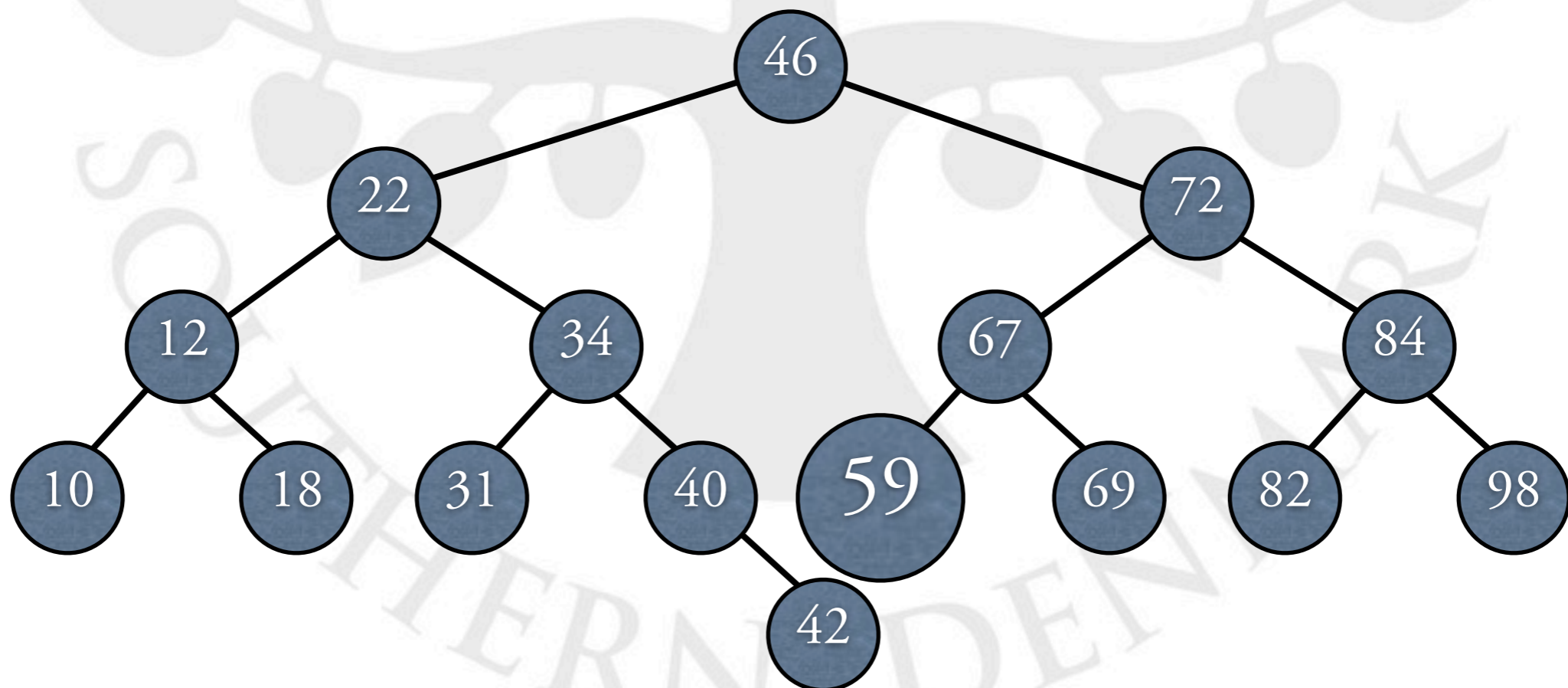
Binære søgetræer

- $prev(e)$ og $next(e)$
 - Let i Array og dobbelt-hægtet liste
- $next(46)$
 - Mindste værdi større end 46



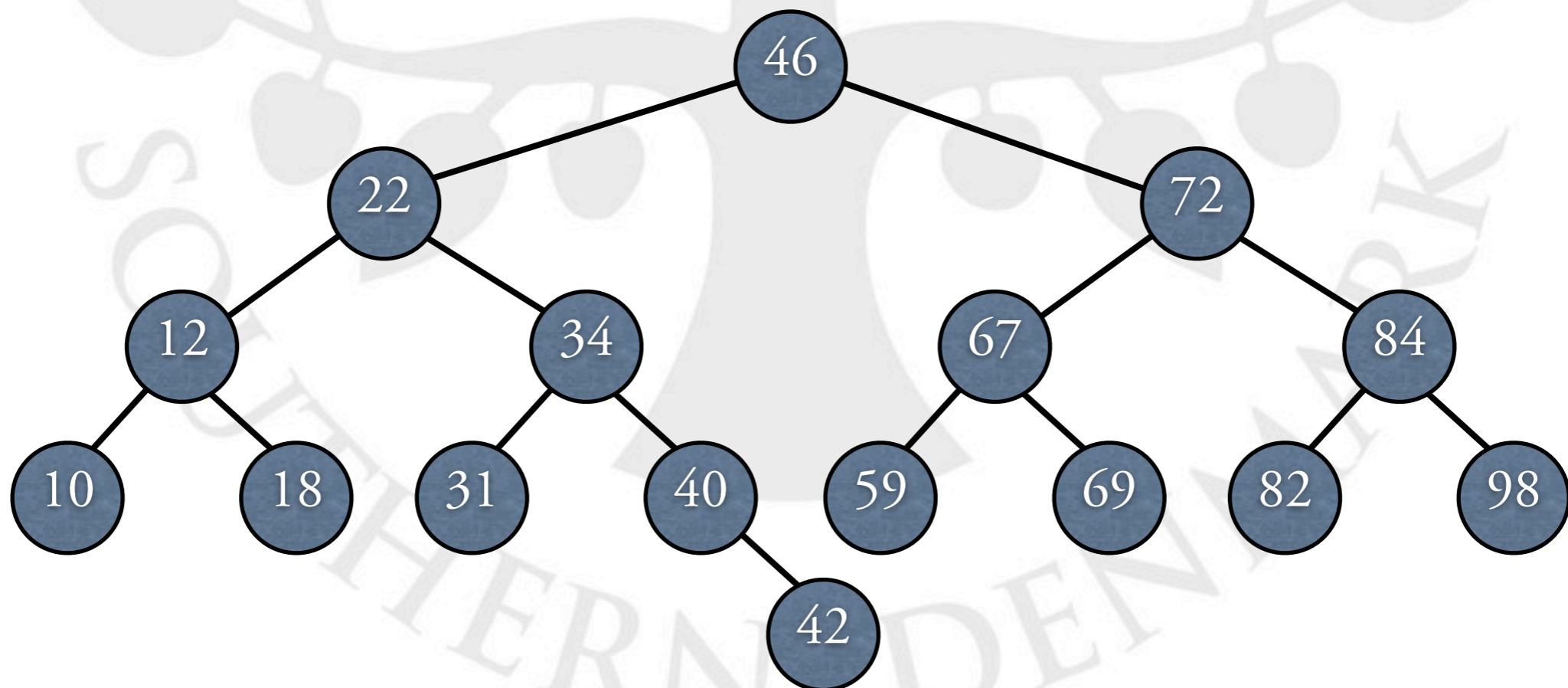
Binære søgetræer

- `prev(e)` og `next(e)`
 - Let i Array og dobbelt-hægtet liste
- `next(46)`
 - Mindste værdi større end 46
 - Knuden længst til venstre i højre undertræ

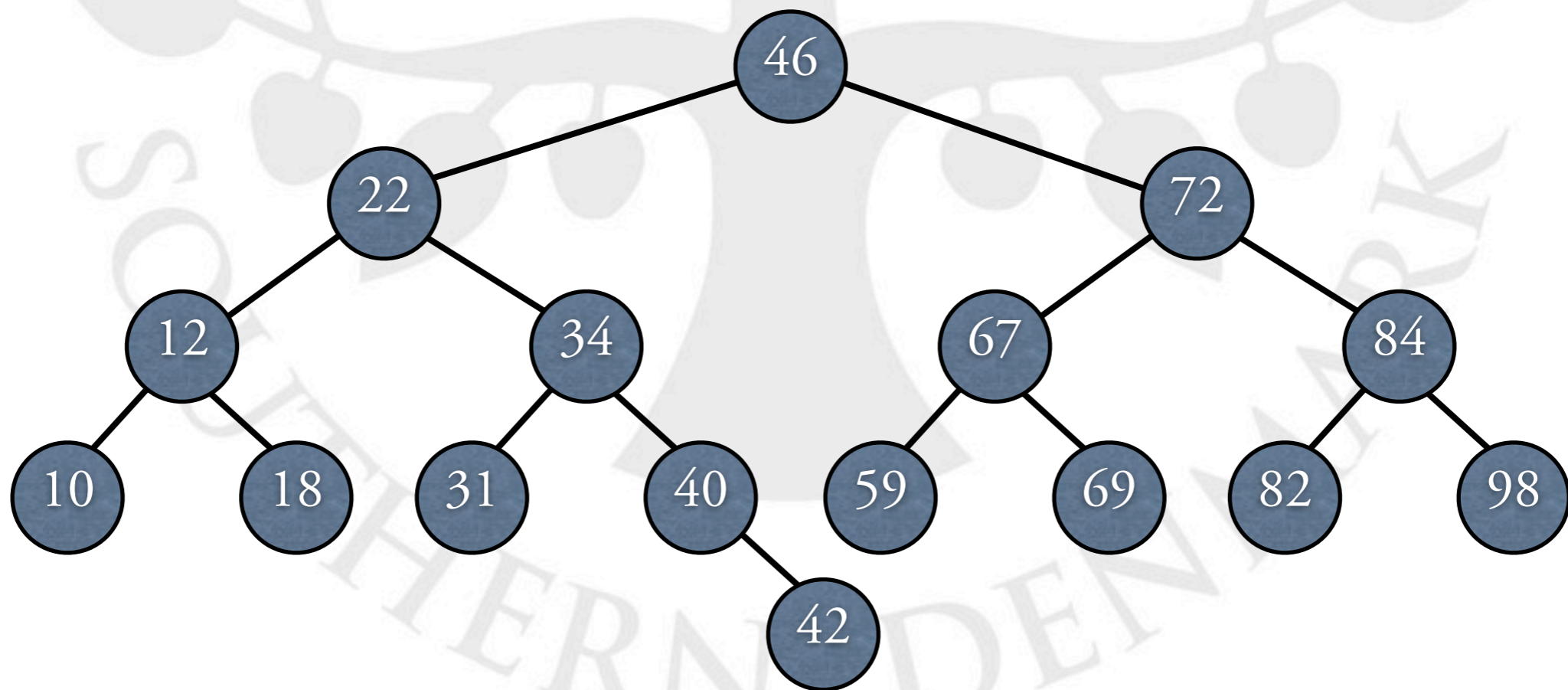


Binære søgetræer

- `prev(e)` og `next(e)`
 - Let i Array og dobbelt-hægtet liste
- `next(46)`
 - Mindste værdi større end 46
 - Knuden længst til venstre i højre undertræ

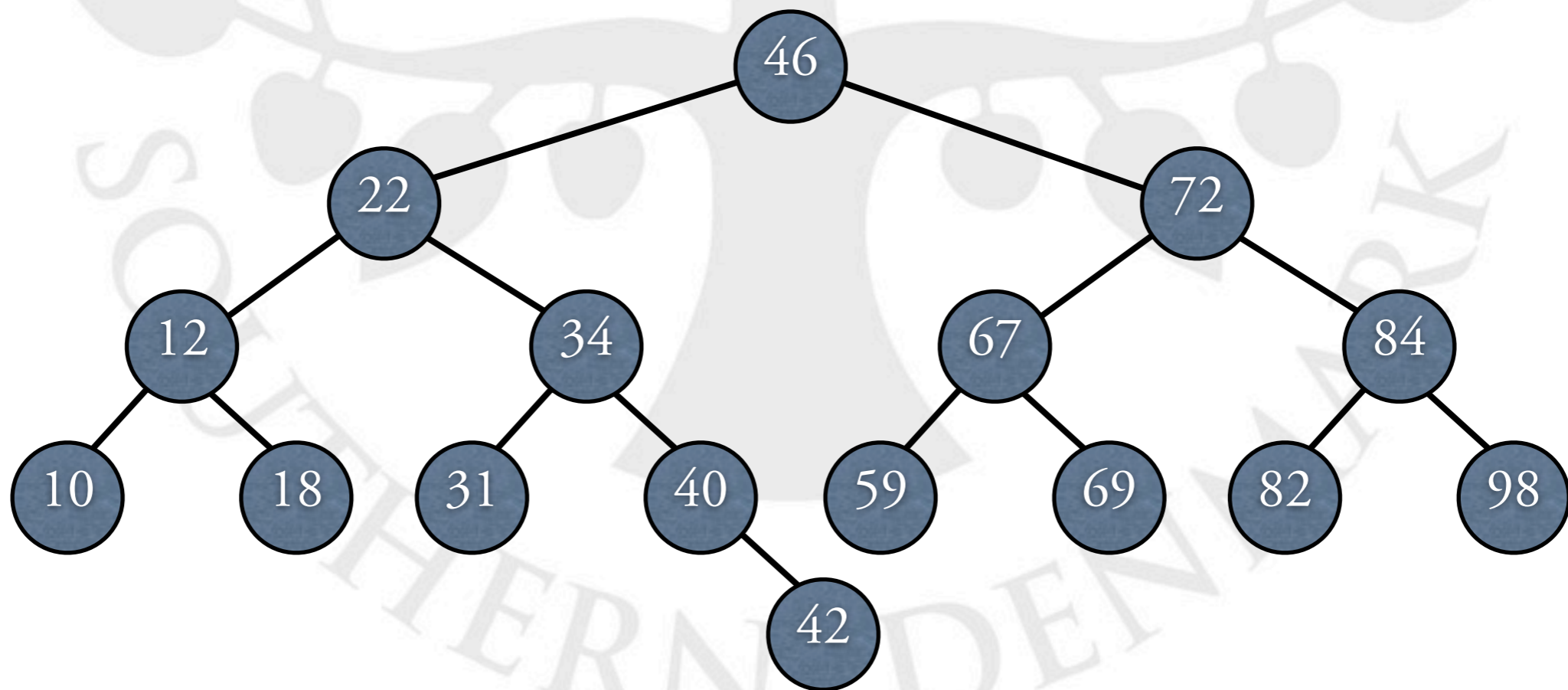


Binære søgetræer



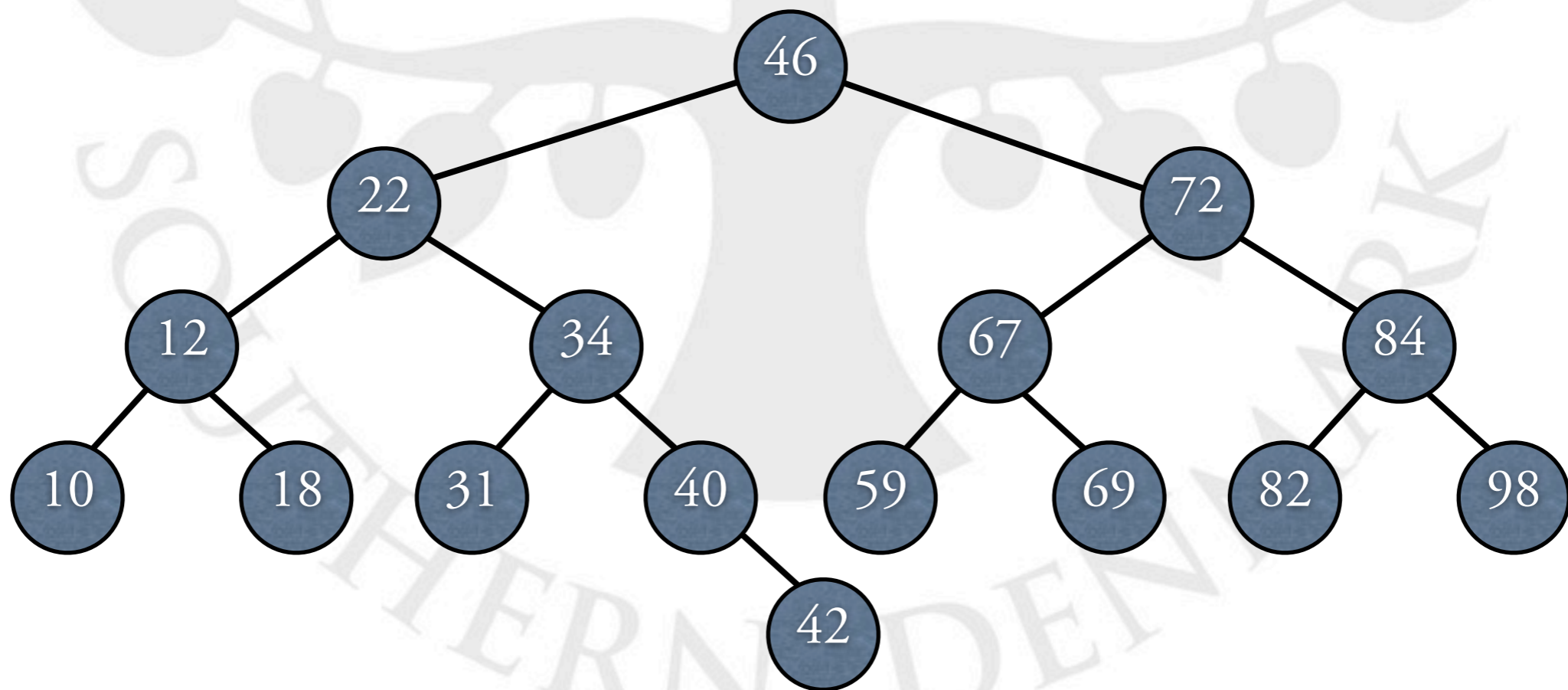
Binære søgetræer

- $\text{prev}(e)$ og $\text{next}(e)$
 - Let i Array og dobbelt-hægtet liste



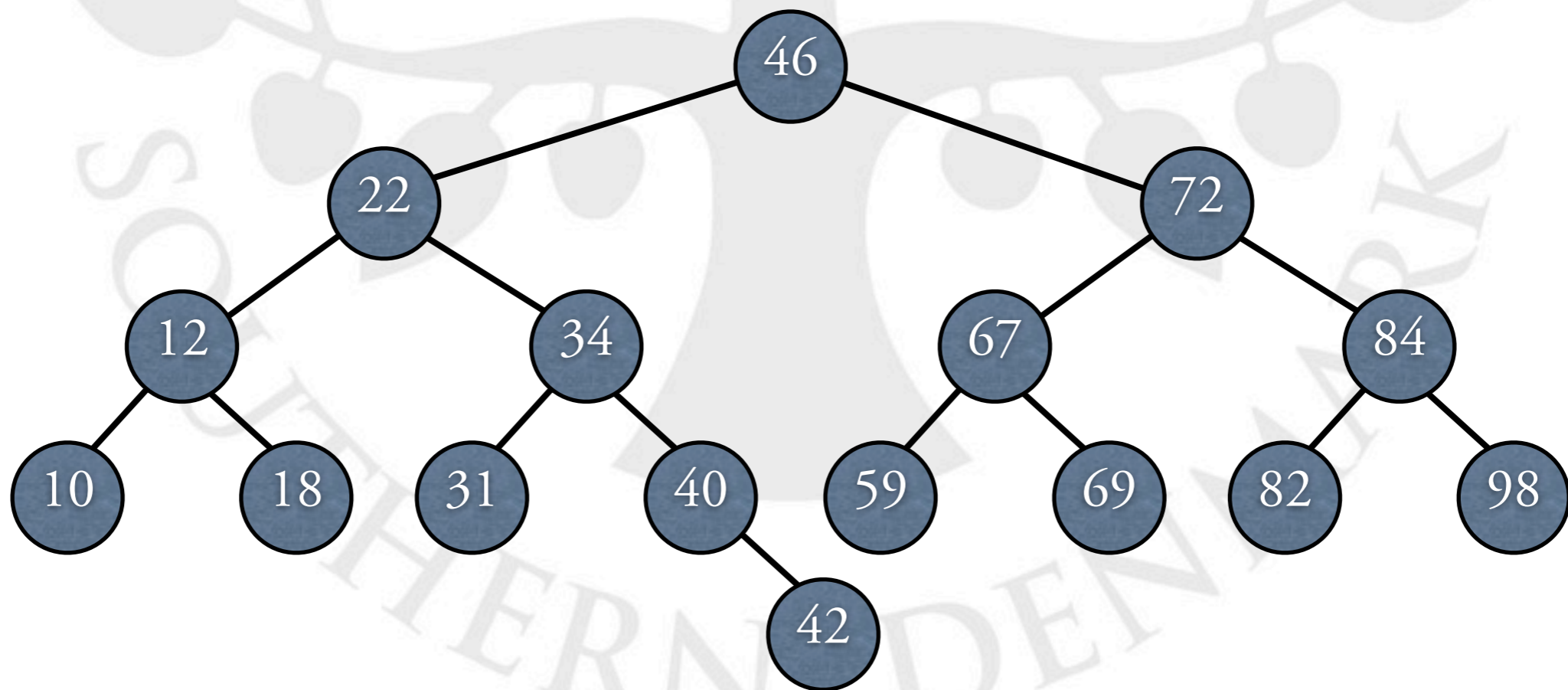
Binære søgetræer

- $\text{prev}(e)$ og $\text{next}(e)$
 - Let i Array og dobbelt-hægtet liste
- $\text{prev}(46)$



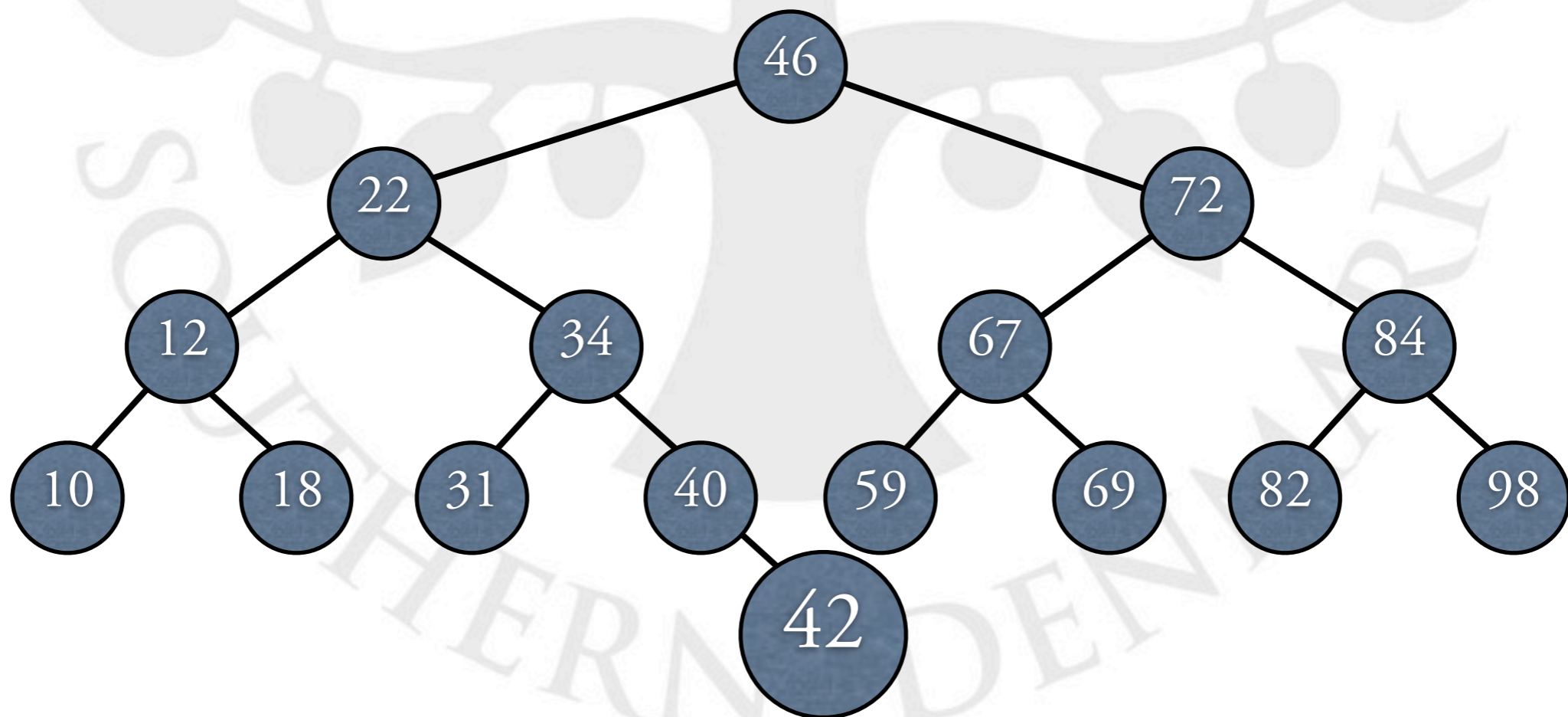
Binære søgetræer

- $prev(e)$ og $next(e)$
 - Let i Array og dobbelt-hægtet liste
- $prev(46)$
 - Største værdi mindre end 46



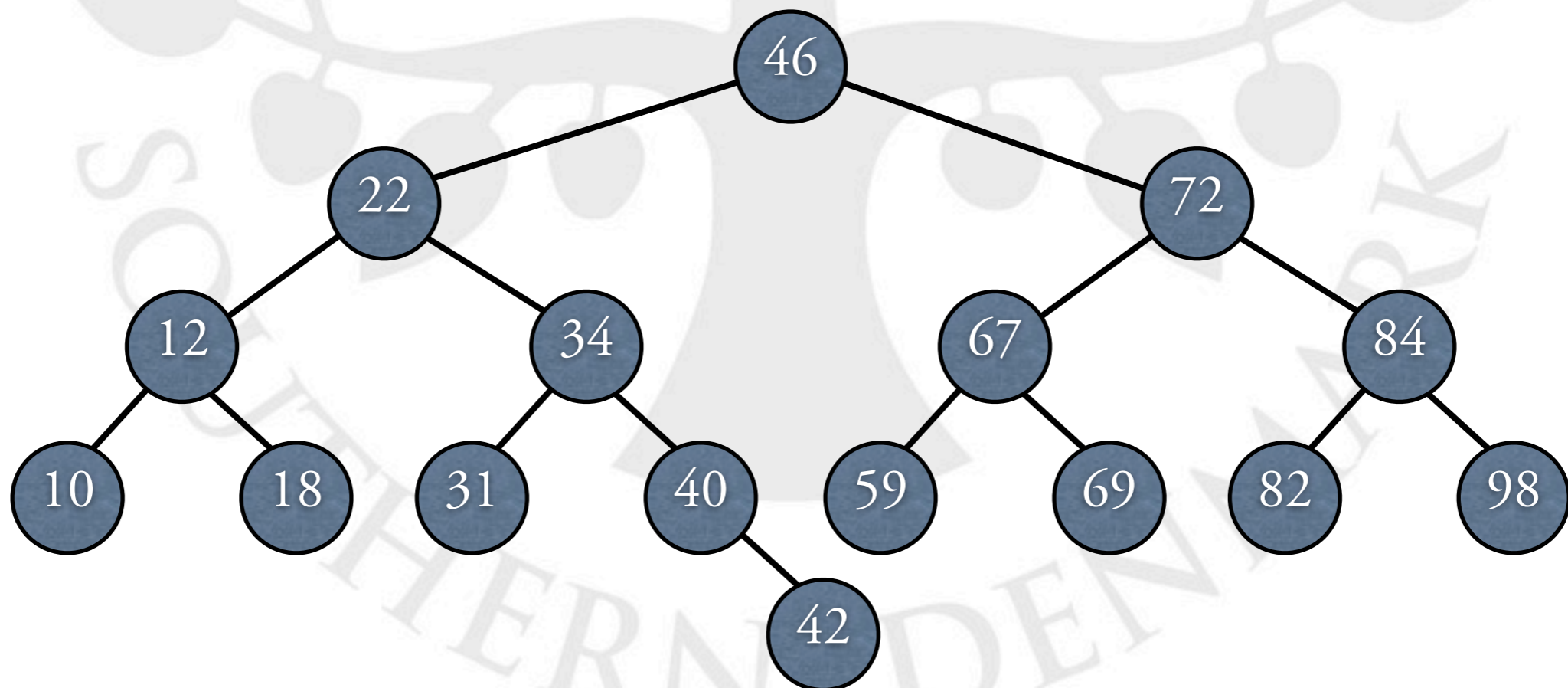
Binære søgetræer

- $\text{prev}(e)$ og $\text{next}(e)$
 - Let i Array og dobbelt-hægtet liste
- $\text{prev}(46)$
 - Største værdi mindre end 46
 - Knuden længst til højre i venstre undertræ

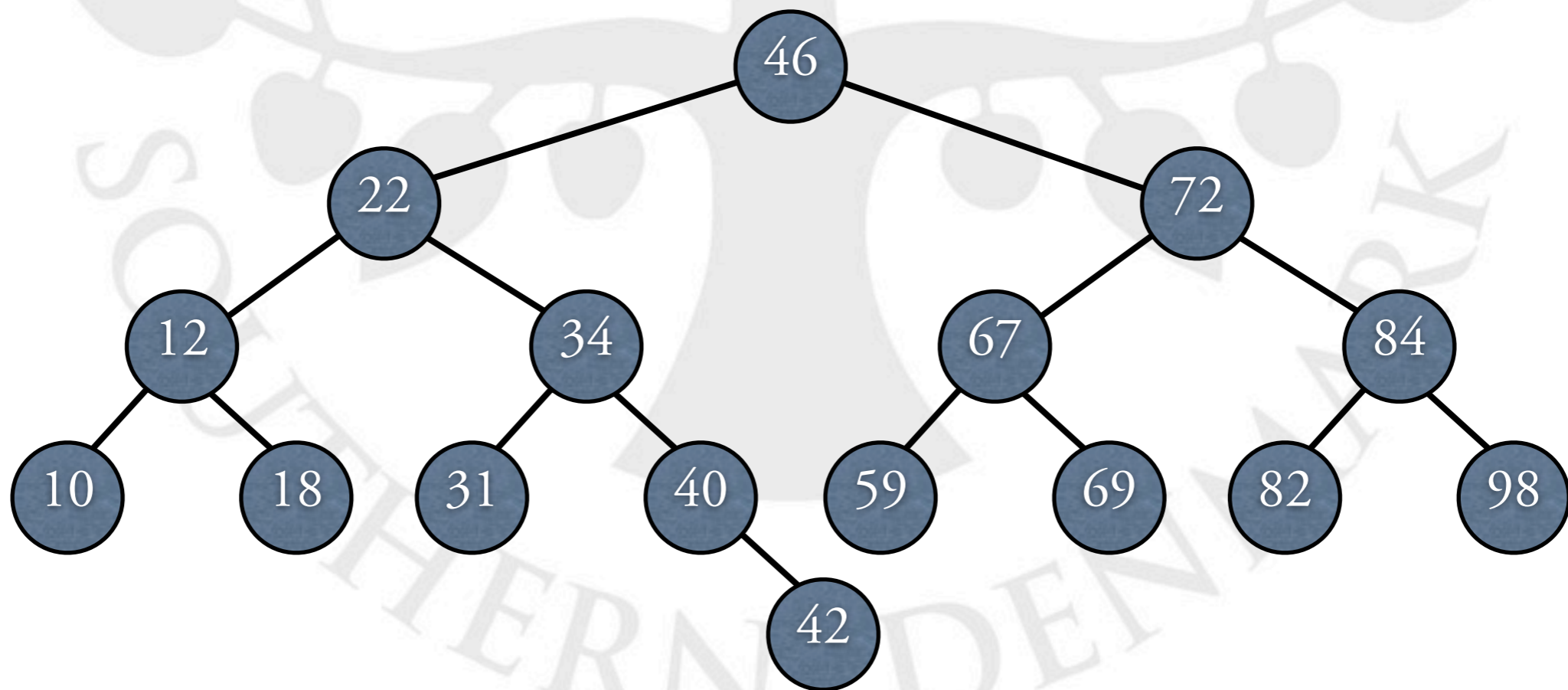


Binære søgetræer

- $prev(e)$ og $next(e)$
 - Let i Array og dobbelt-hægtet liste
- $prev(46)$
 - Største værdi mindre end 46
 - Knuden længst til højre i venstre undertræ

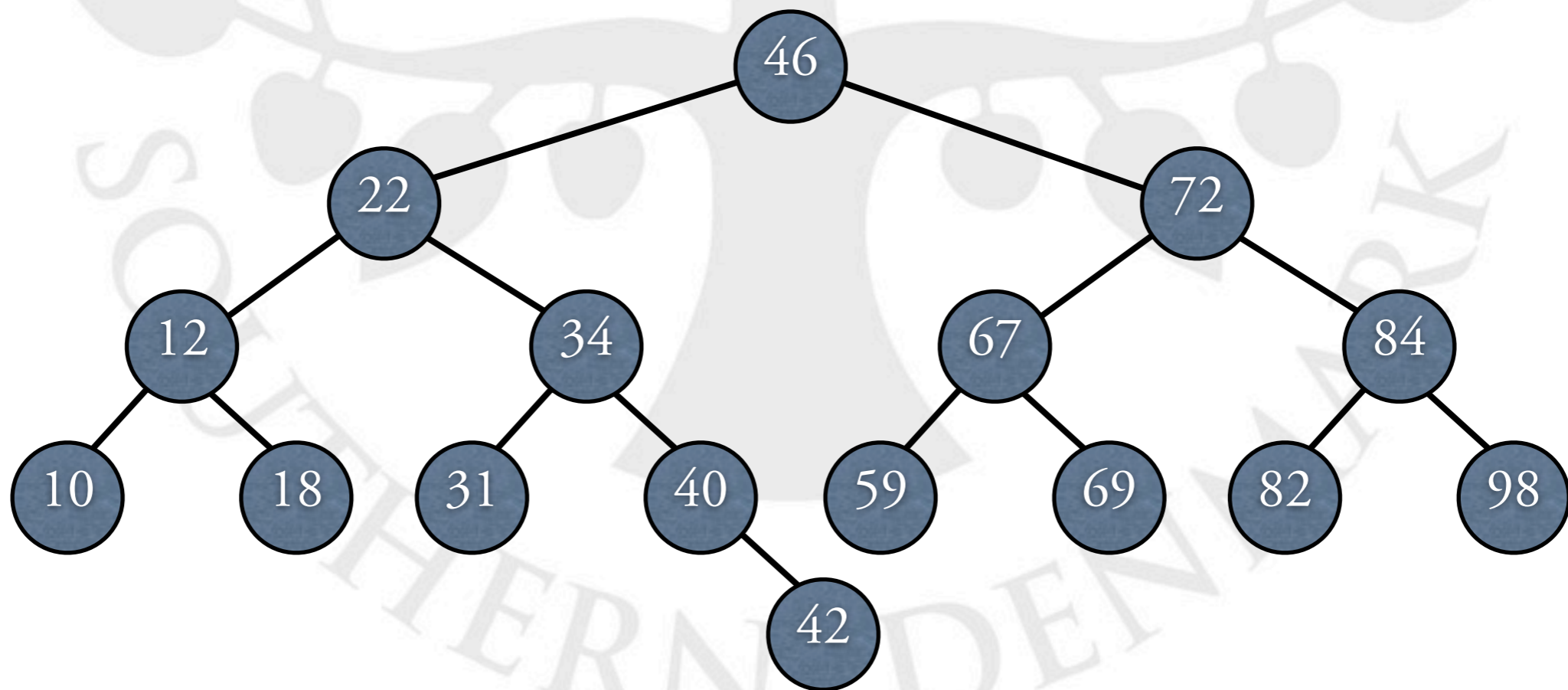


Binære søgetræer



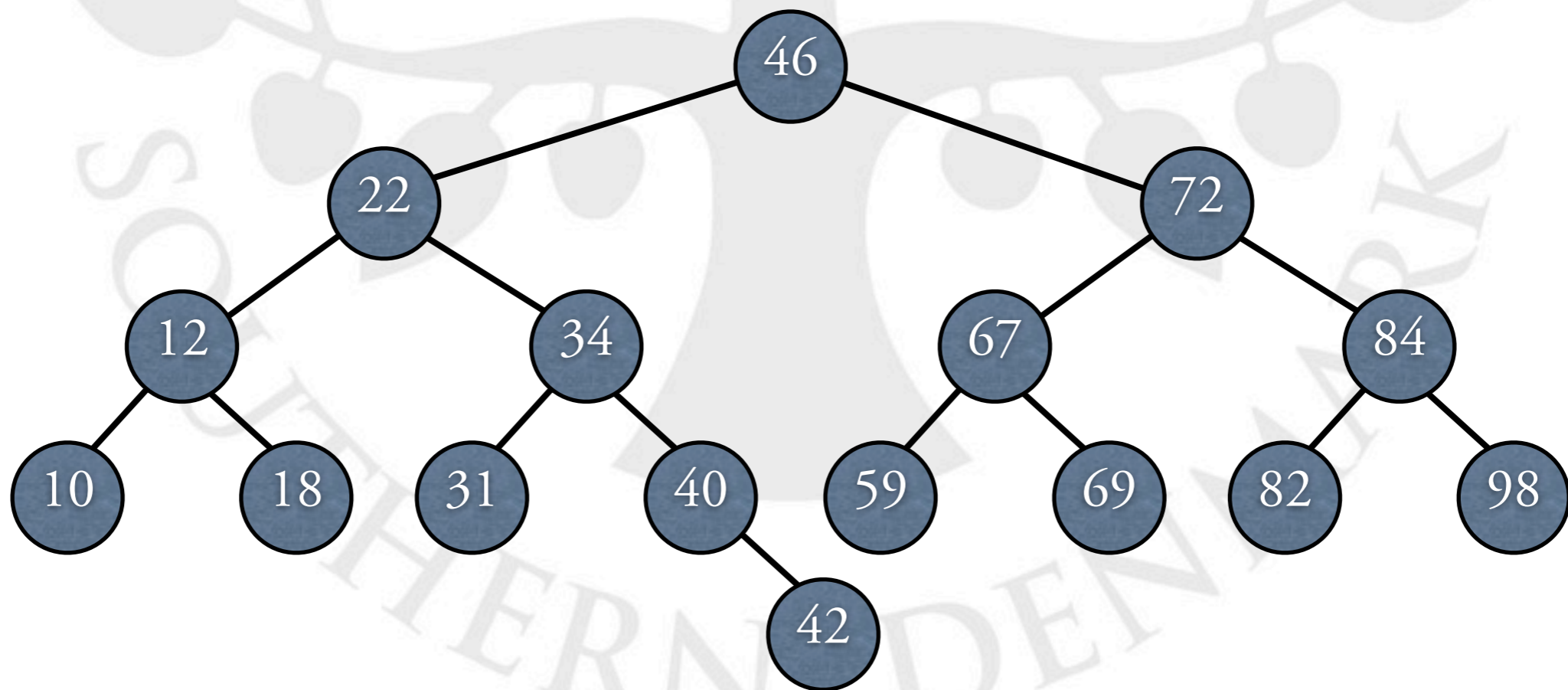
Binære søgetræer

- Hvad med sletninger?



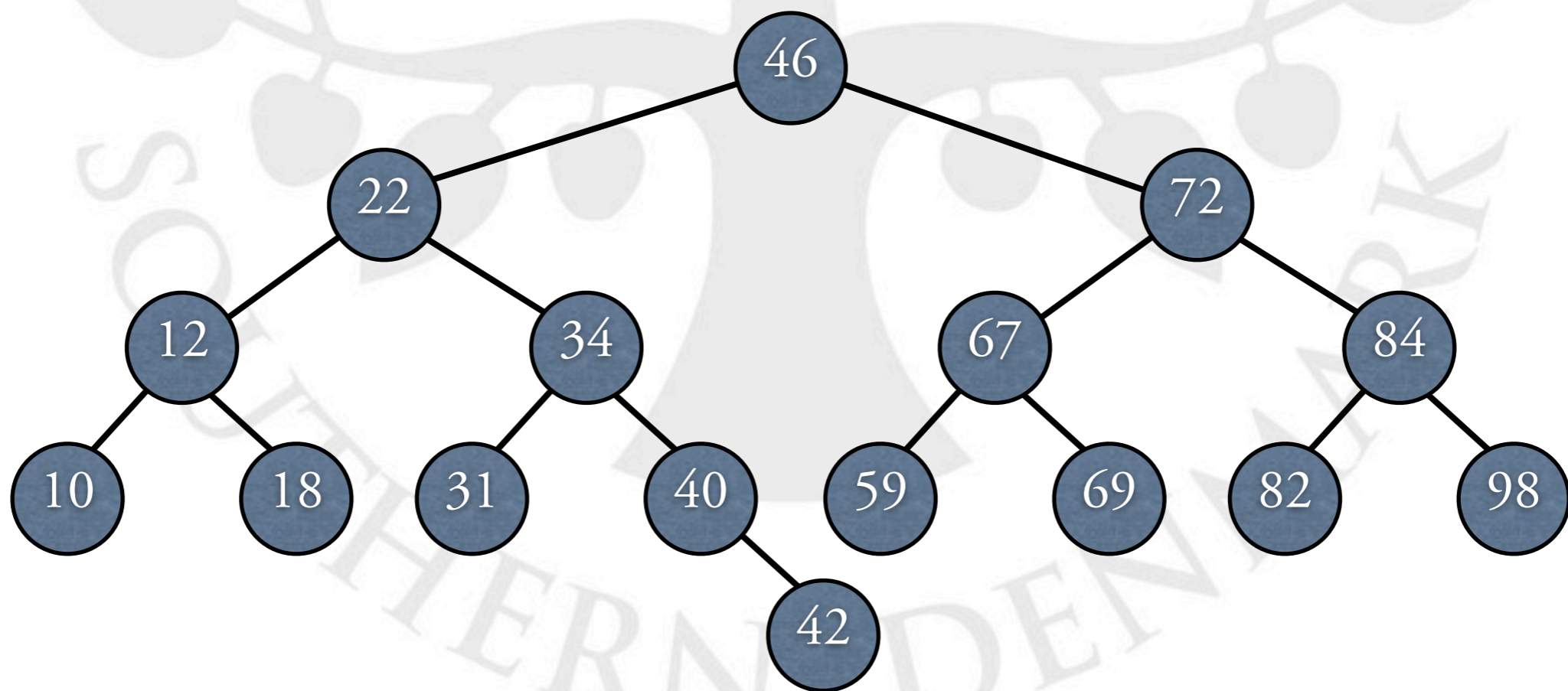
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde



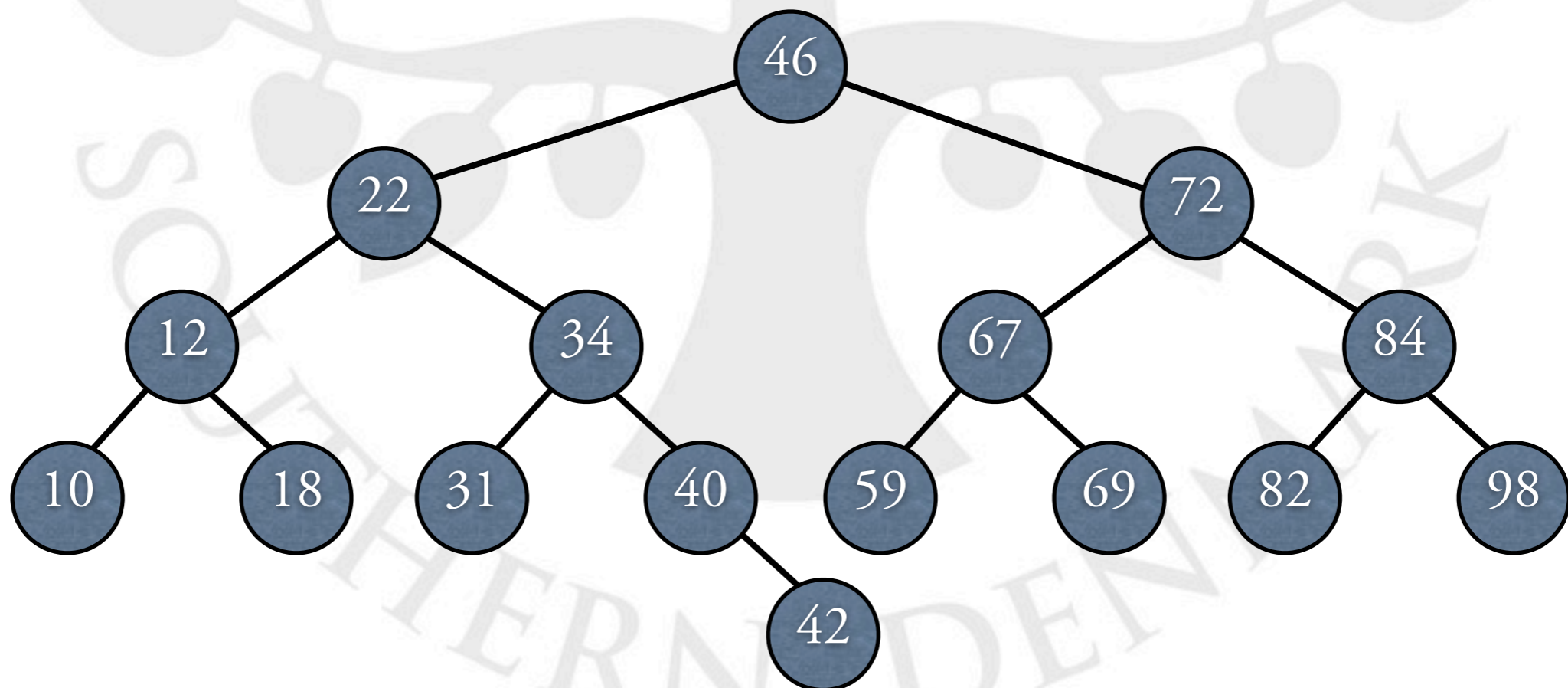
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af blad



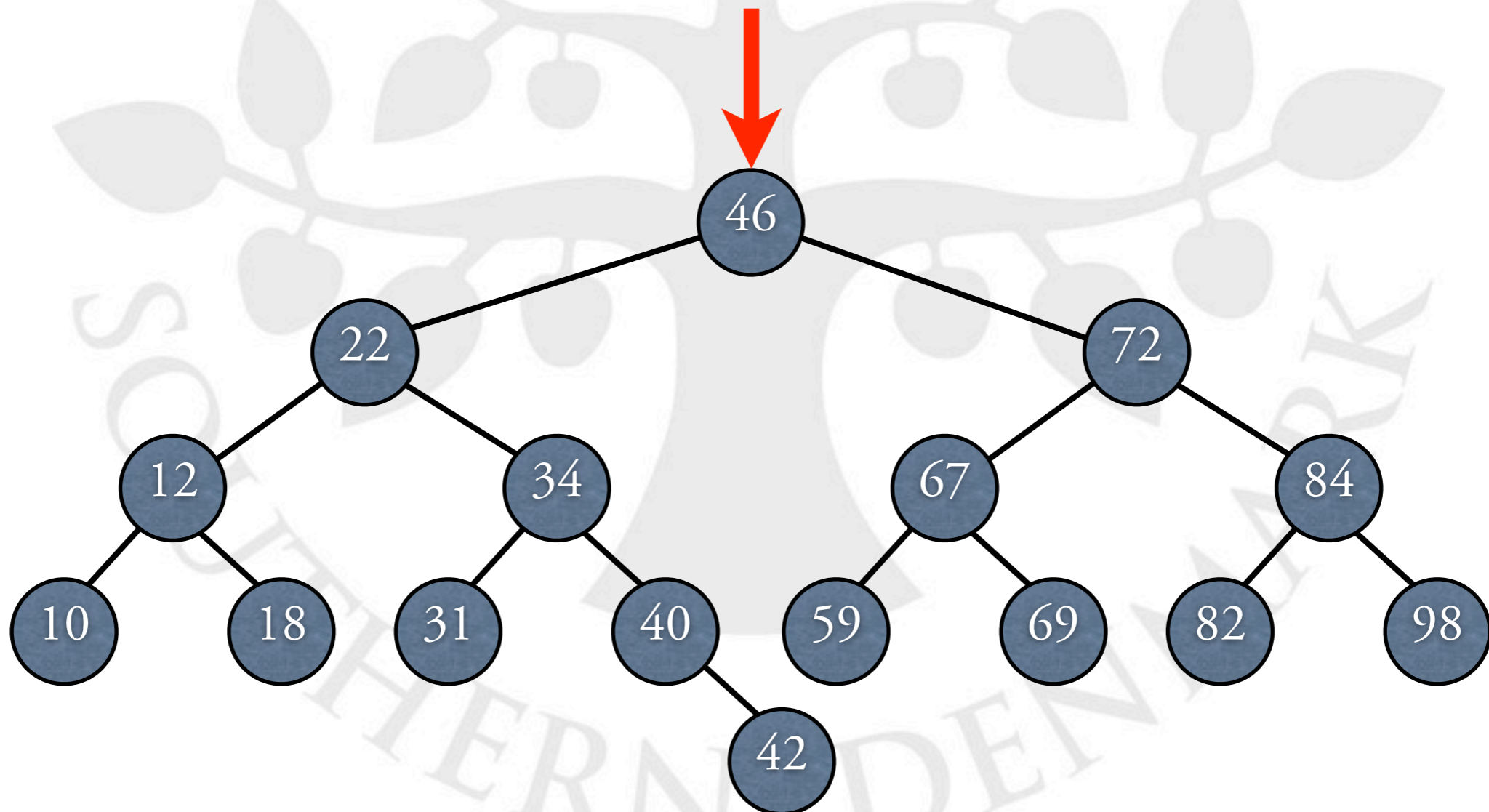
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af blad
 - Remove(42)



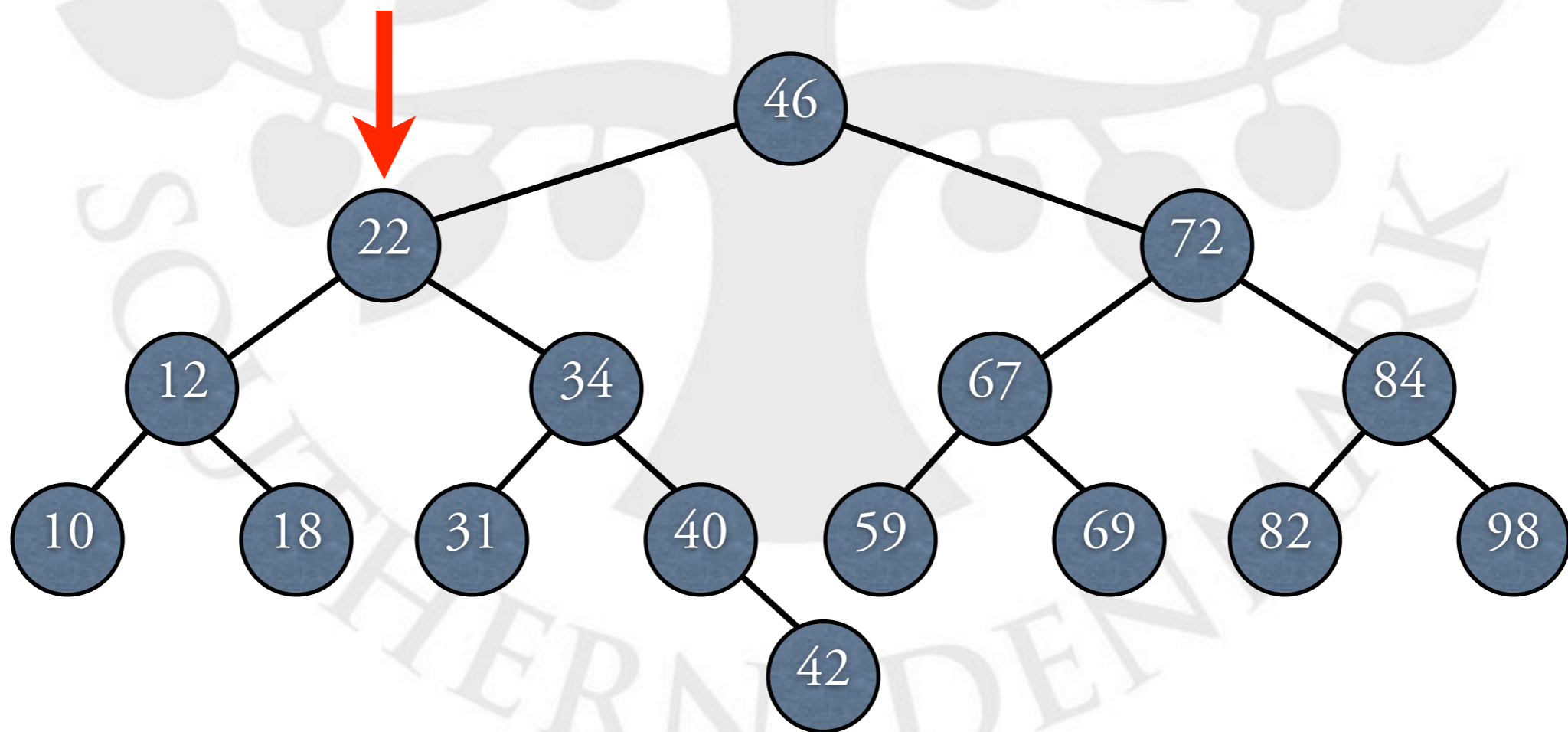
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af blad
 - Remove(42)



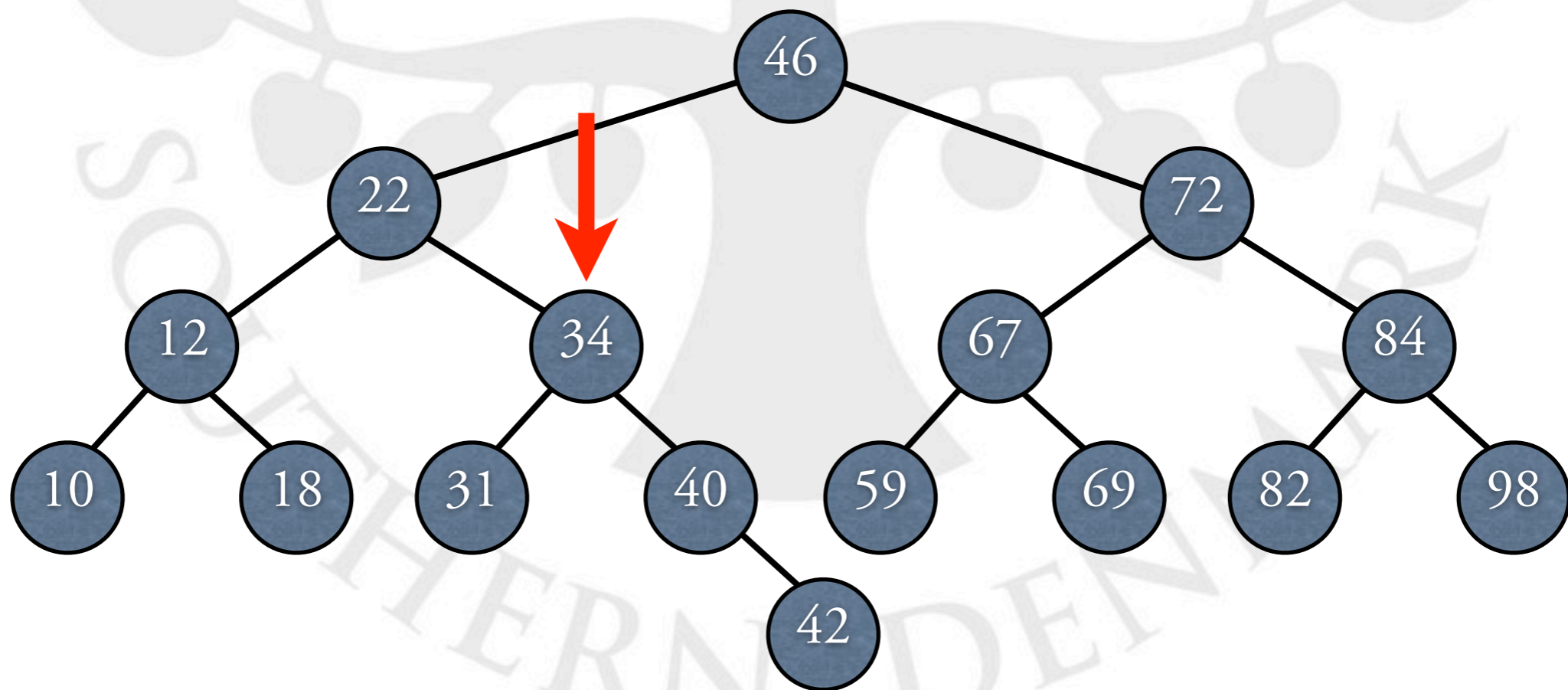
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af blad
 - Remove(42)



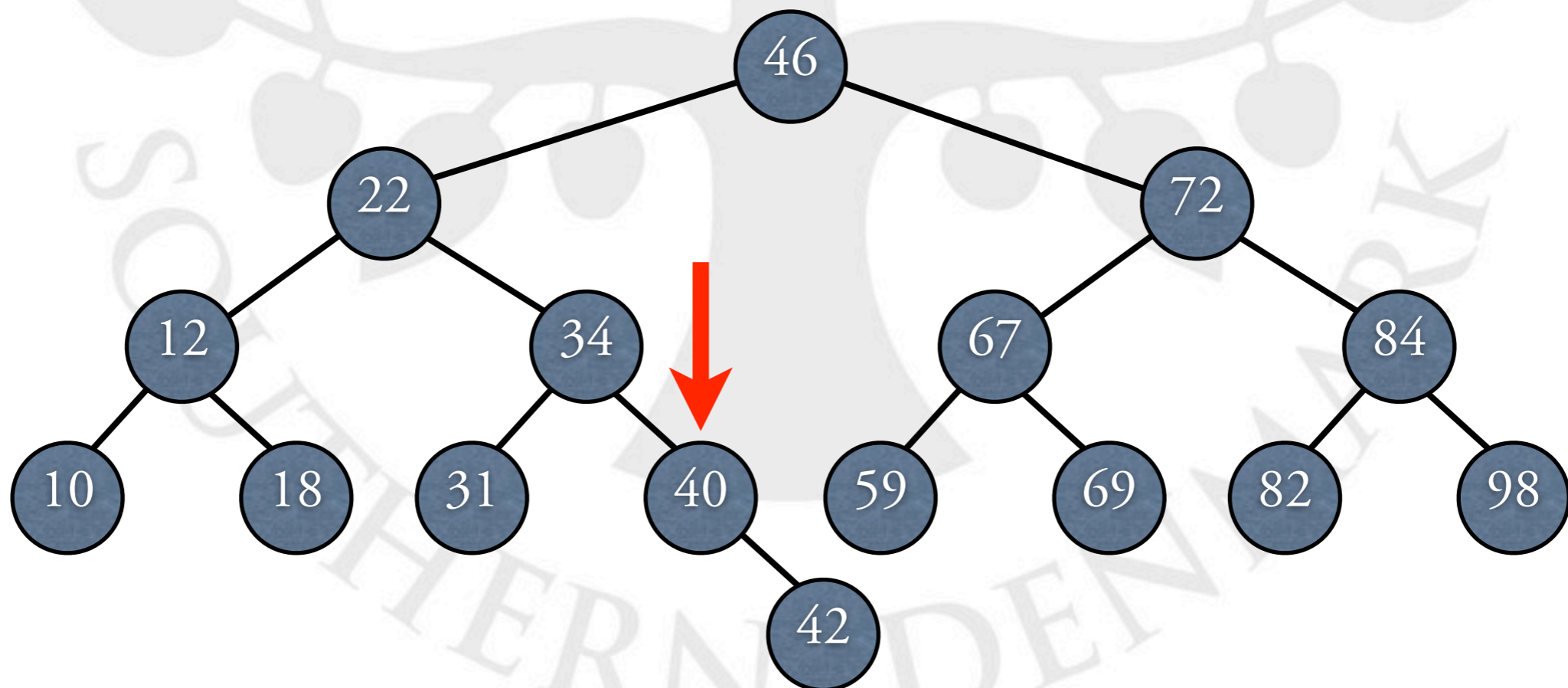
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af blad
 - Remove(42)



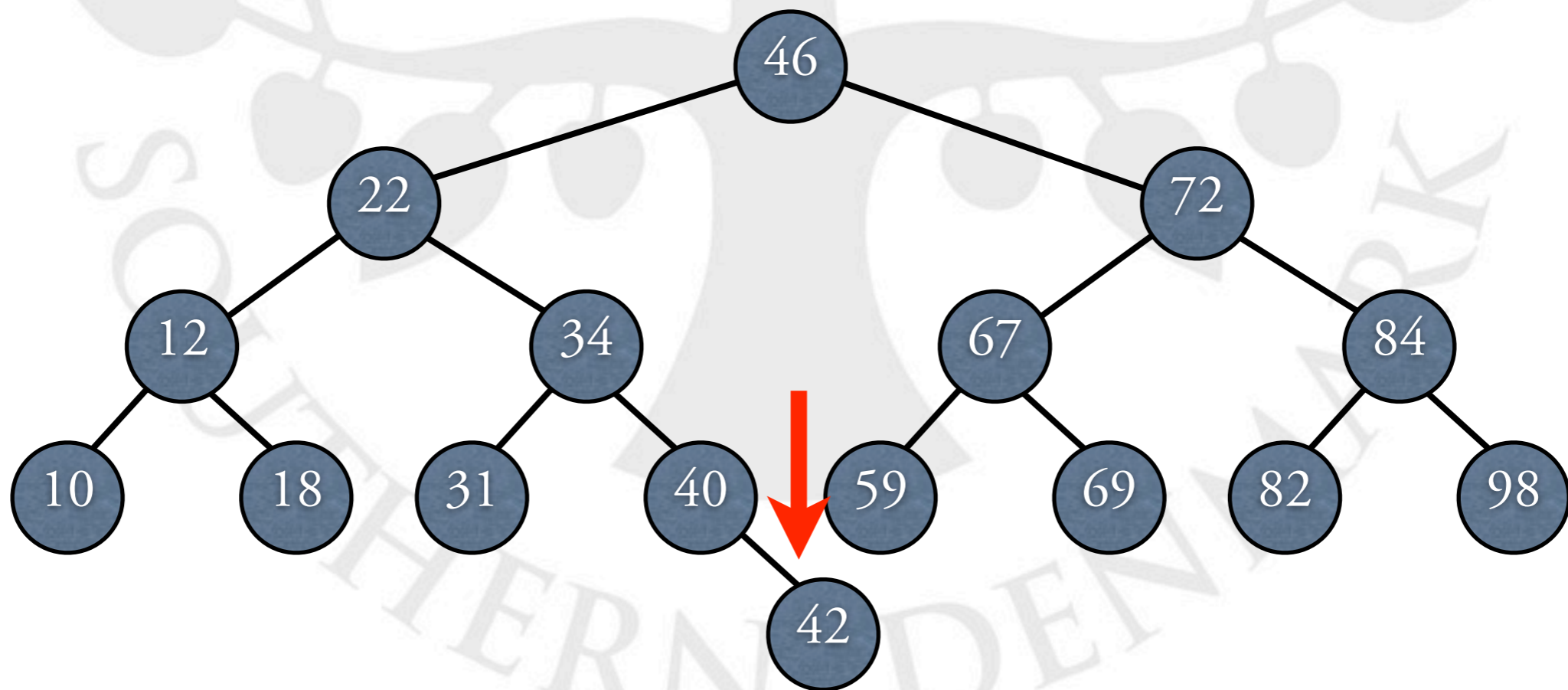
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af blad
 - Remove(42)



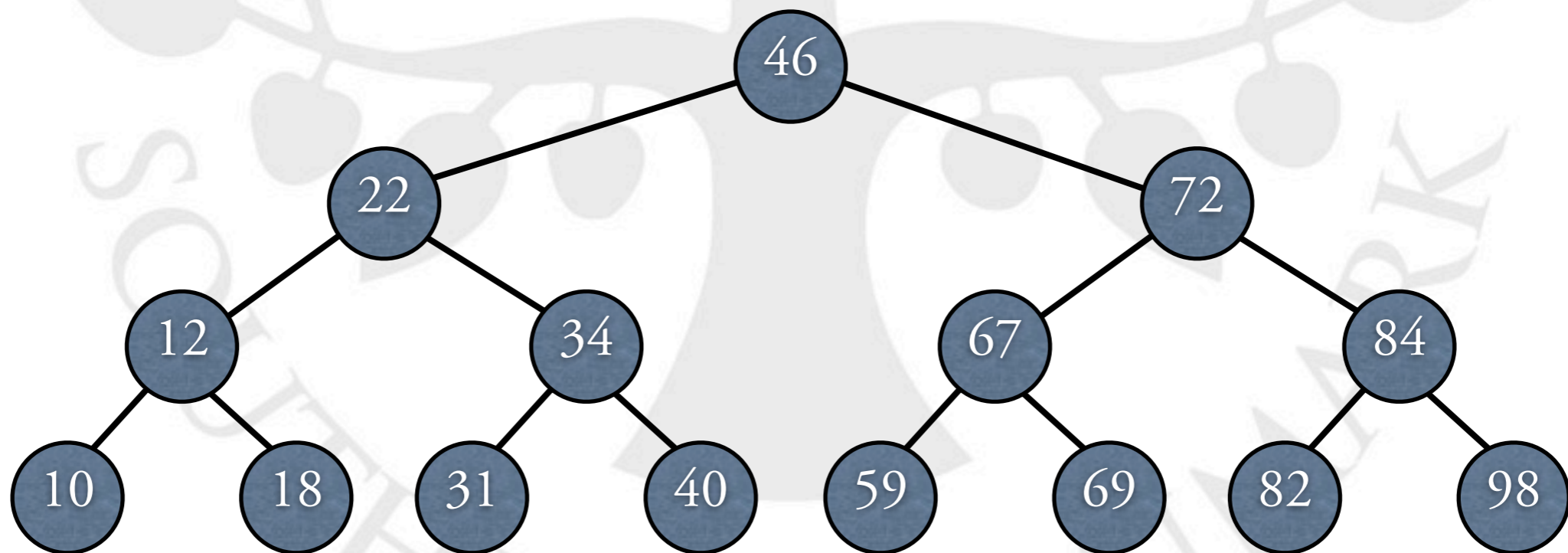
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af blad
 - Remove(42)

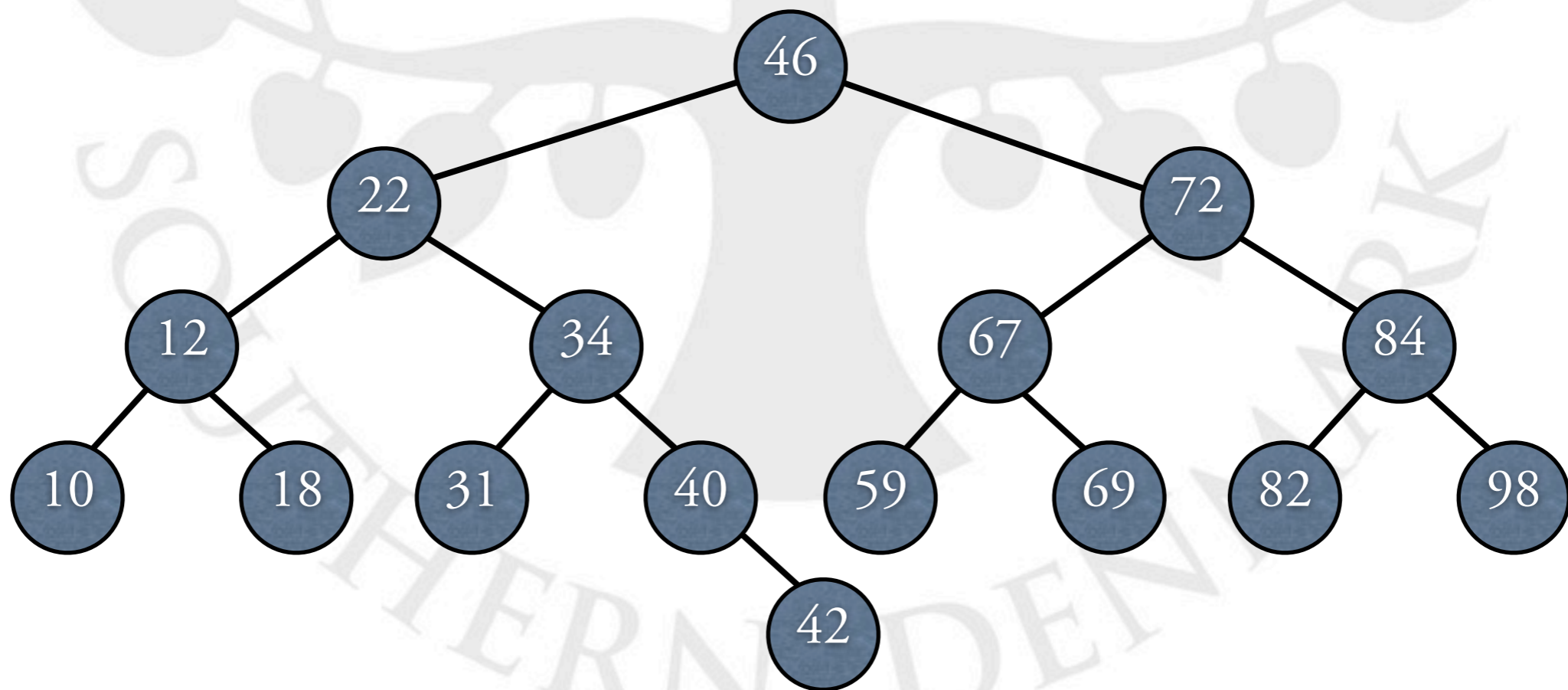


Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af blad
 - Remove(42)

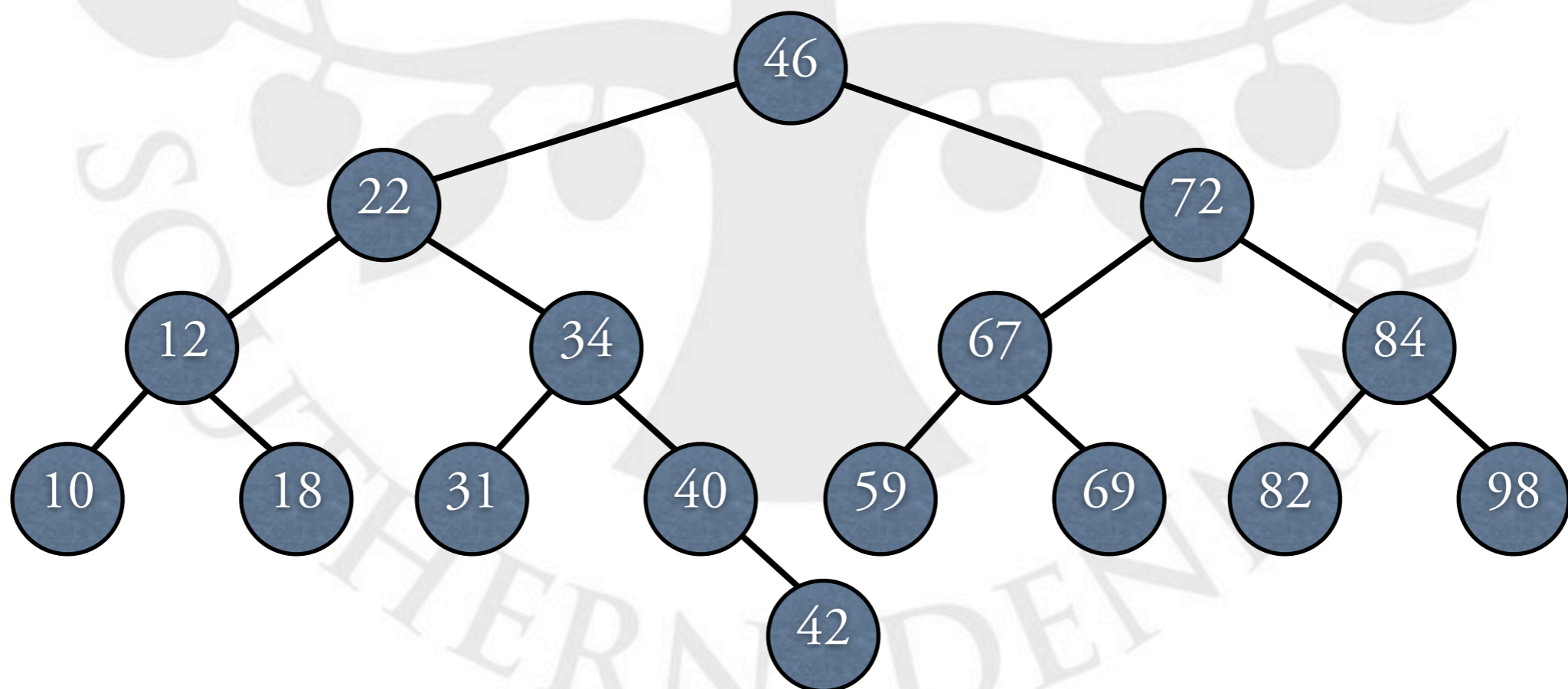


Binære søgetræer



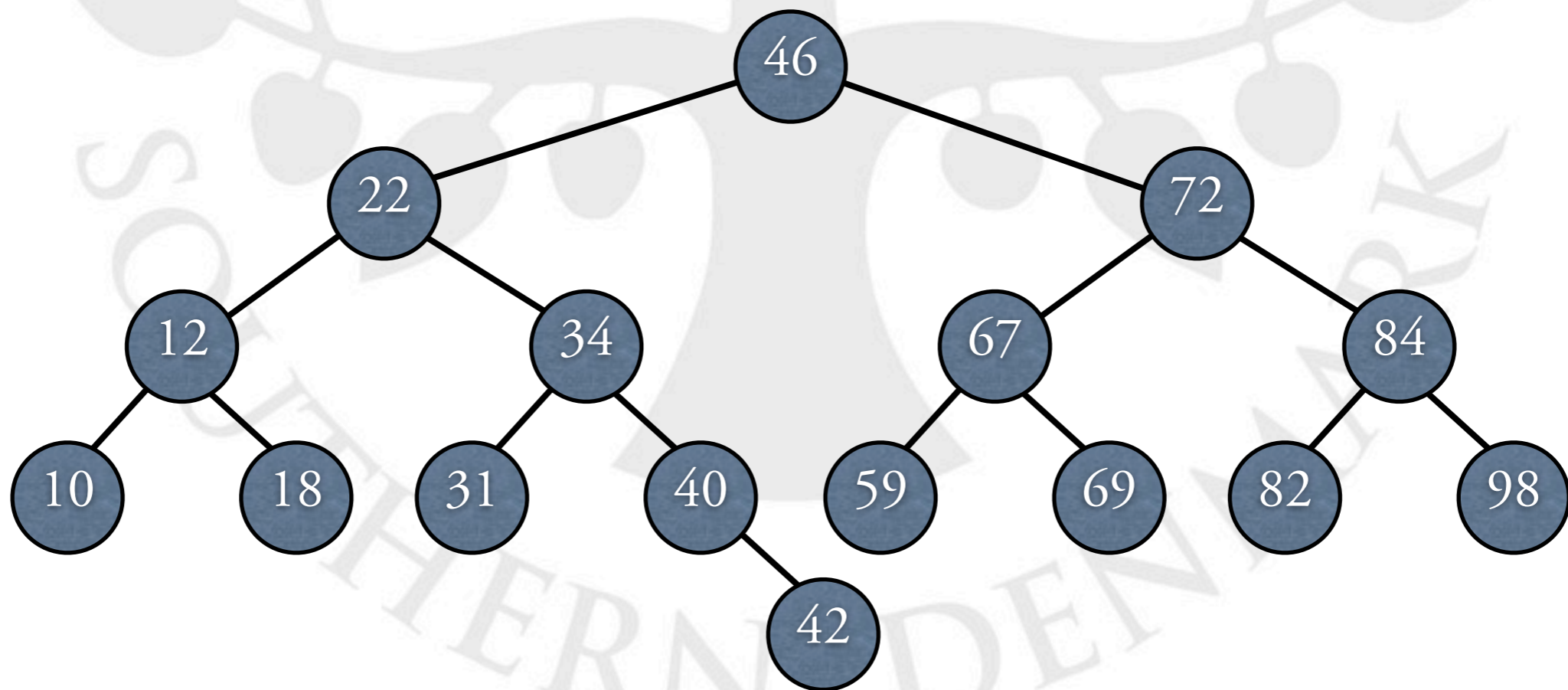
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde



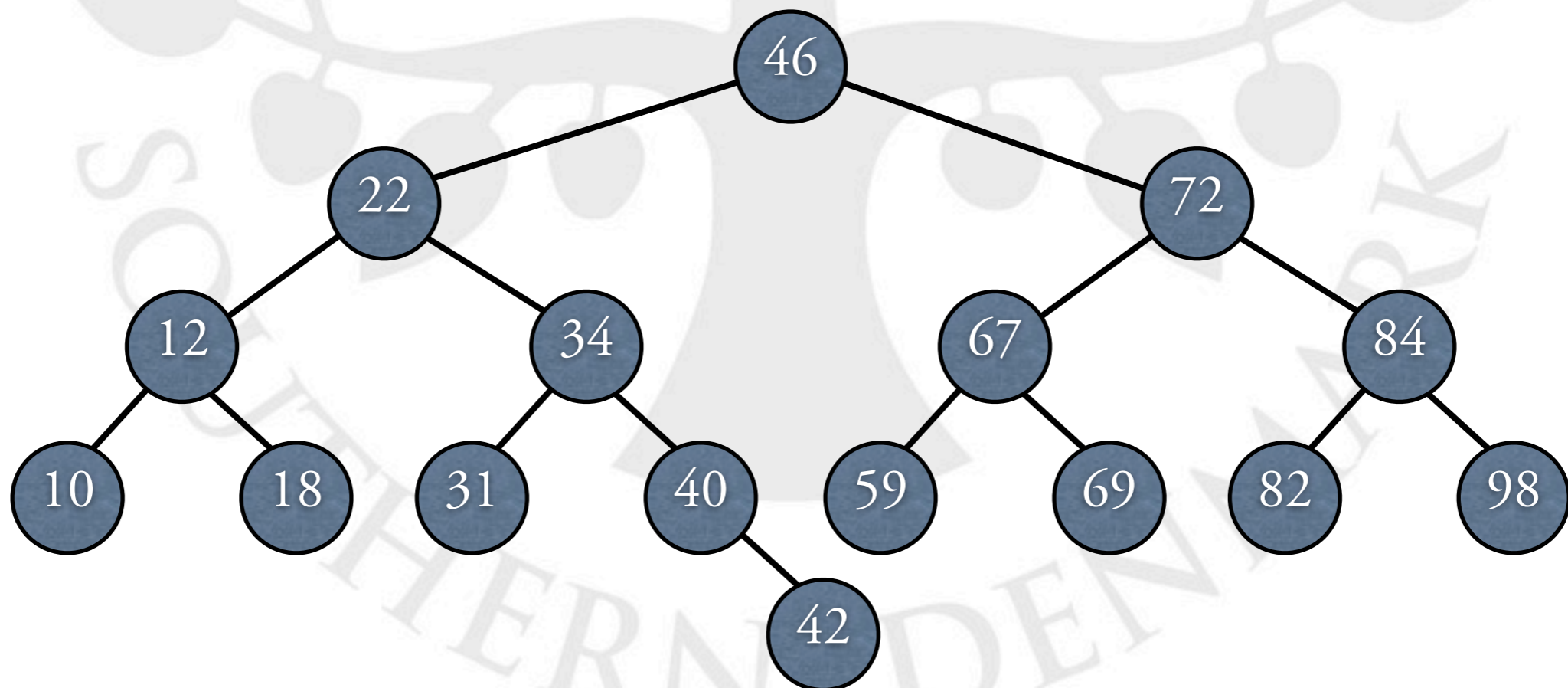
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med ét barn



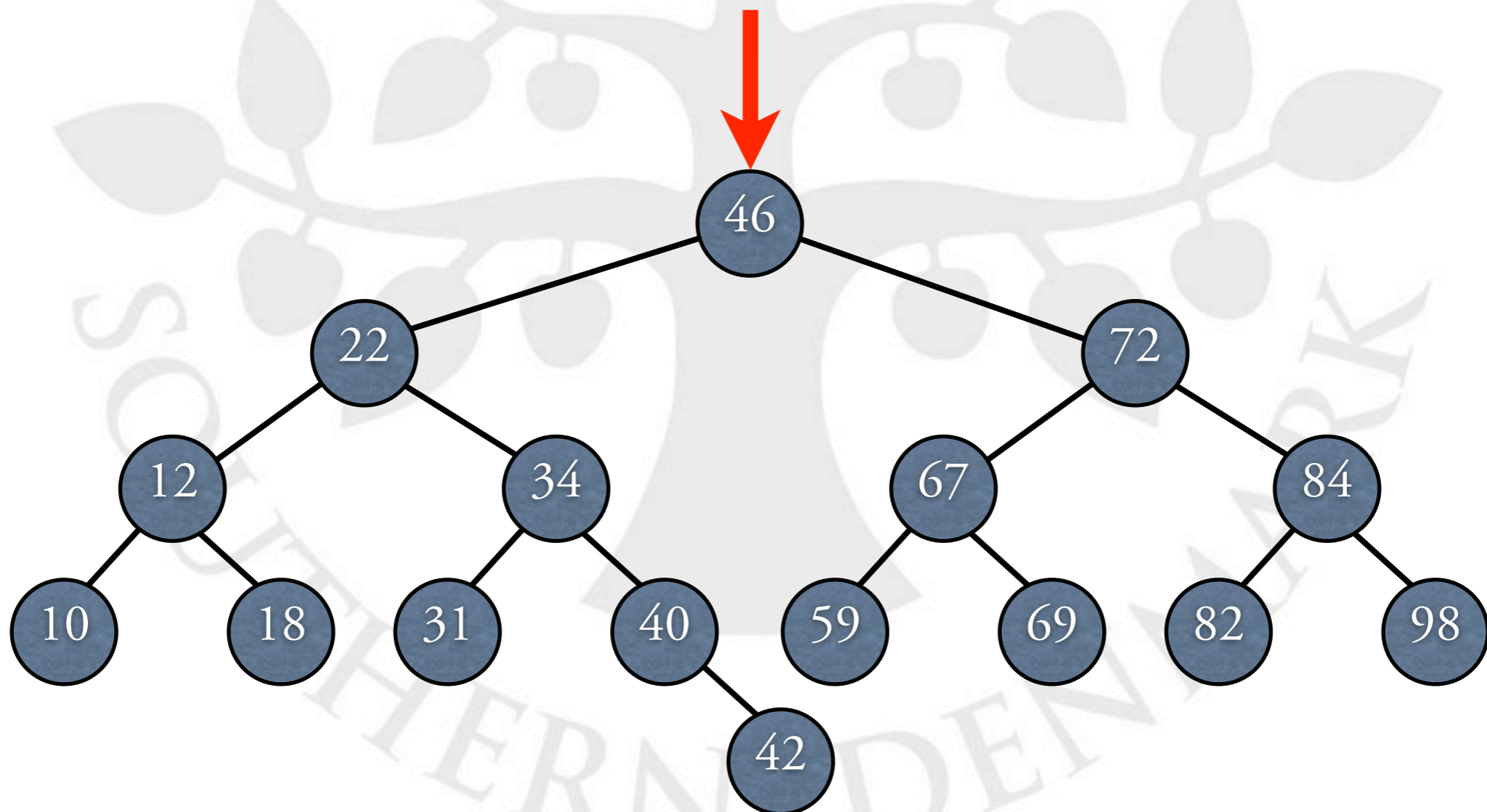
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med ét barn
 - Remove(40)



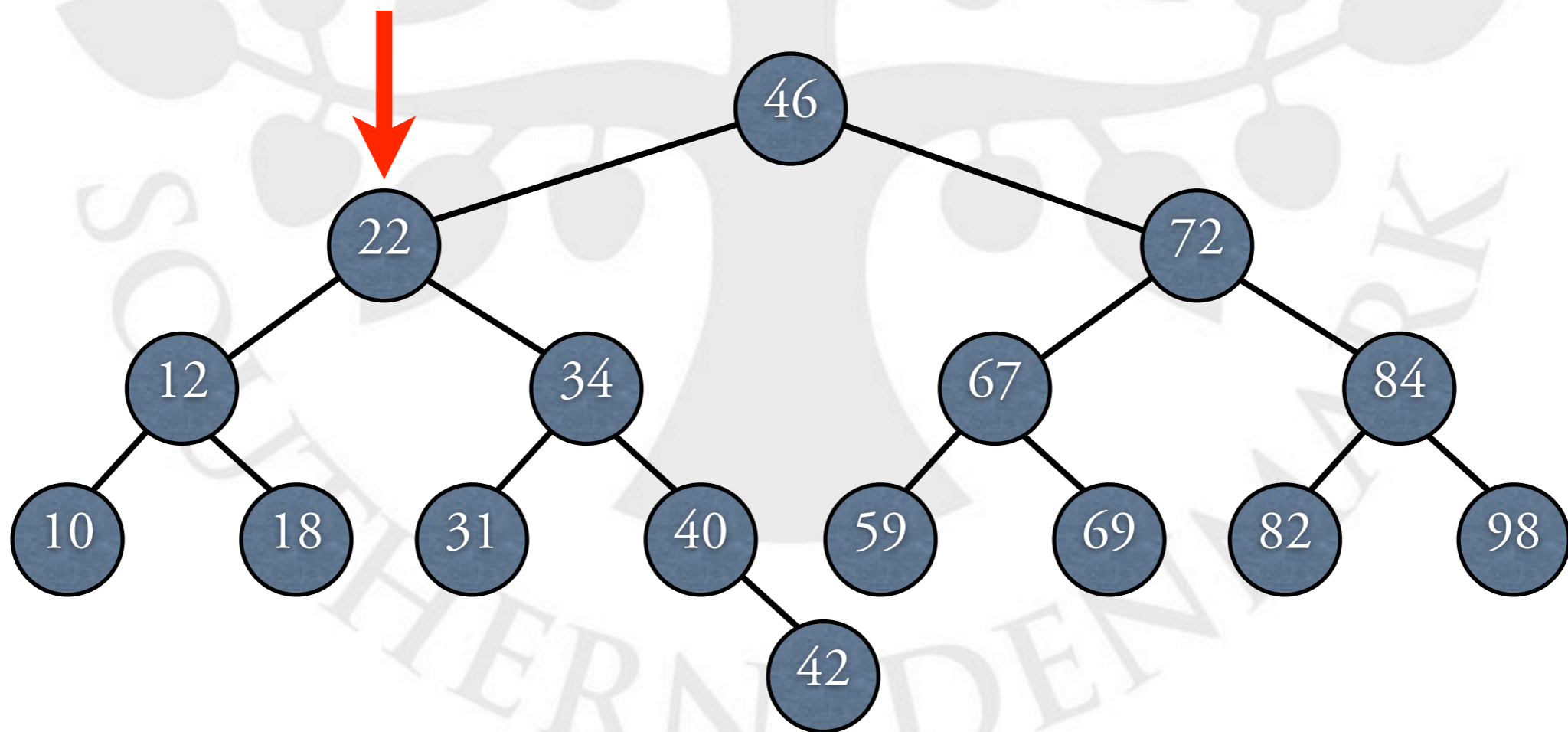
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med ét barn
 - Remove(40)



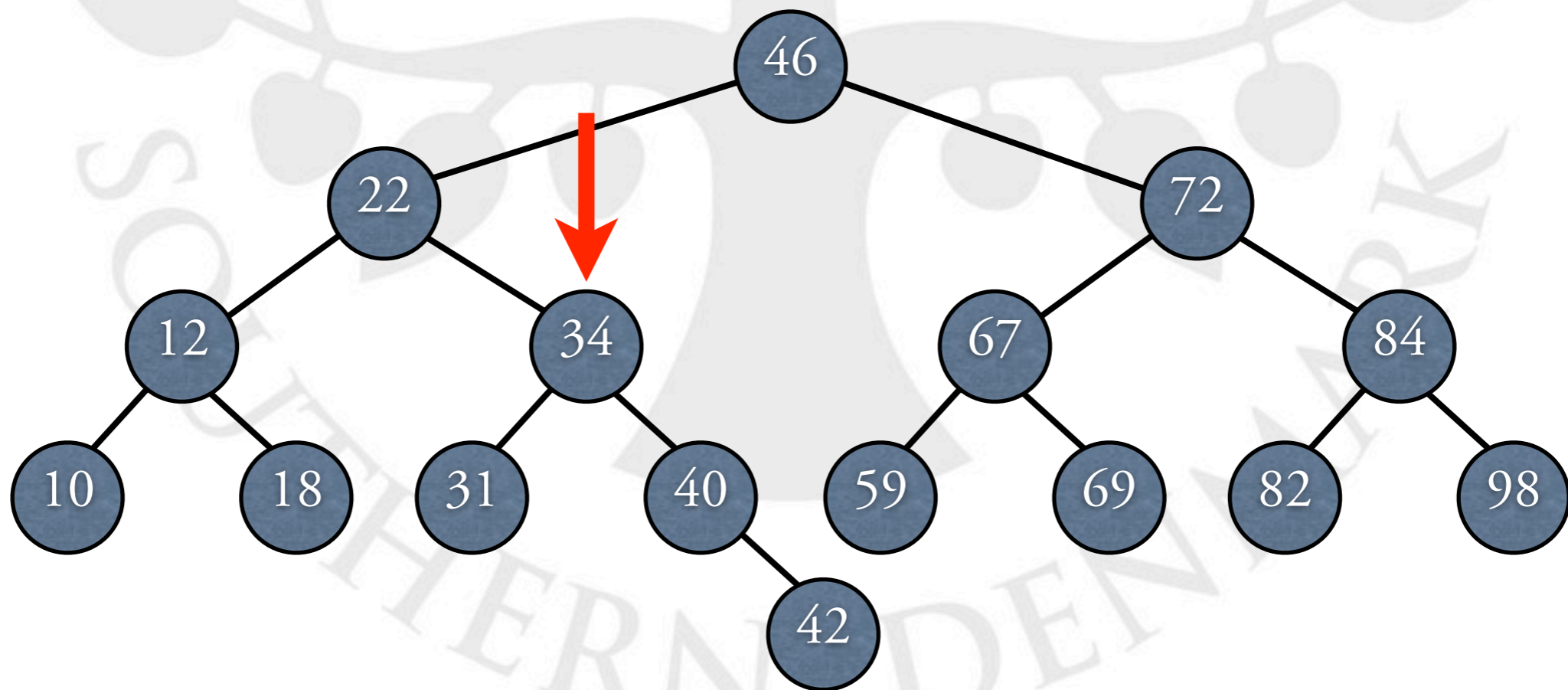
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med ét barn
 - Remove(40)



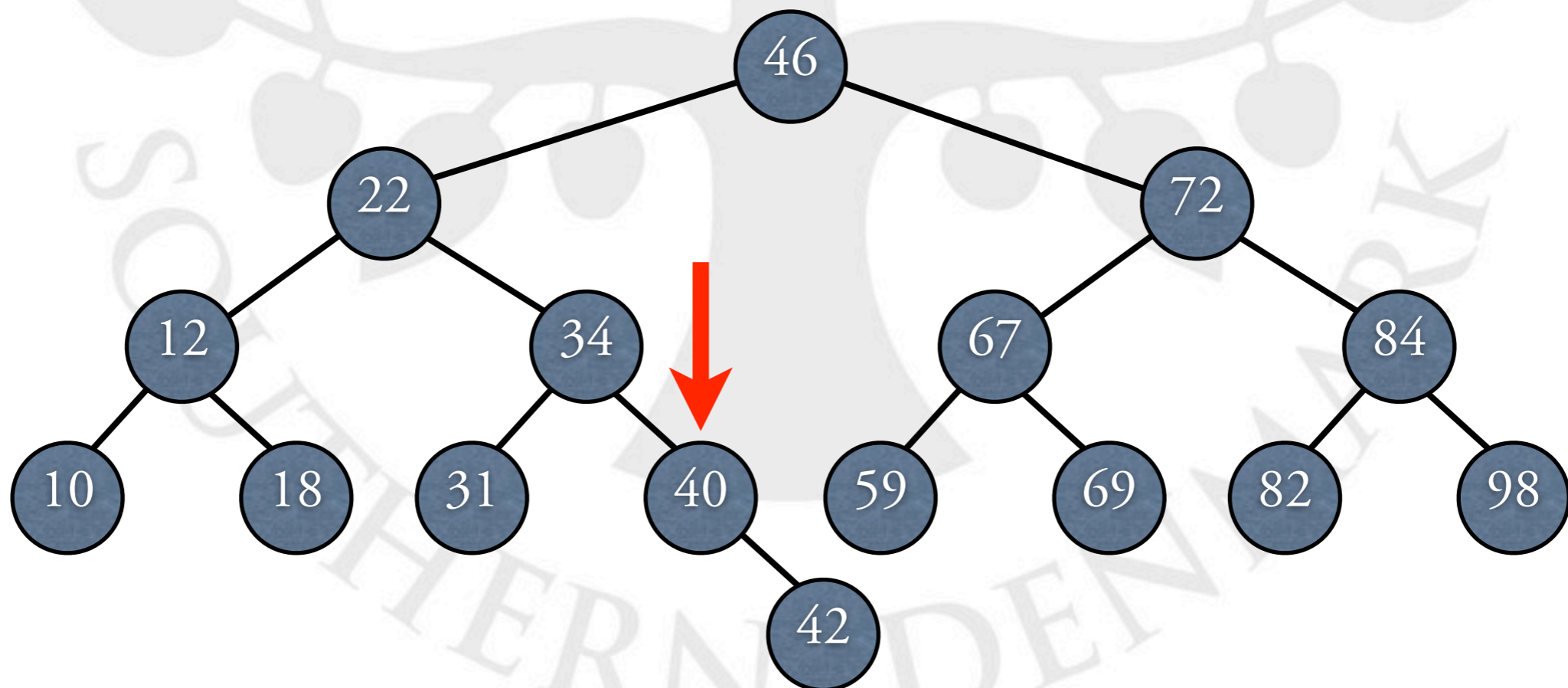
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med ét barn
 - Remove(40)



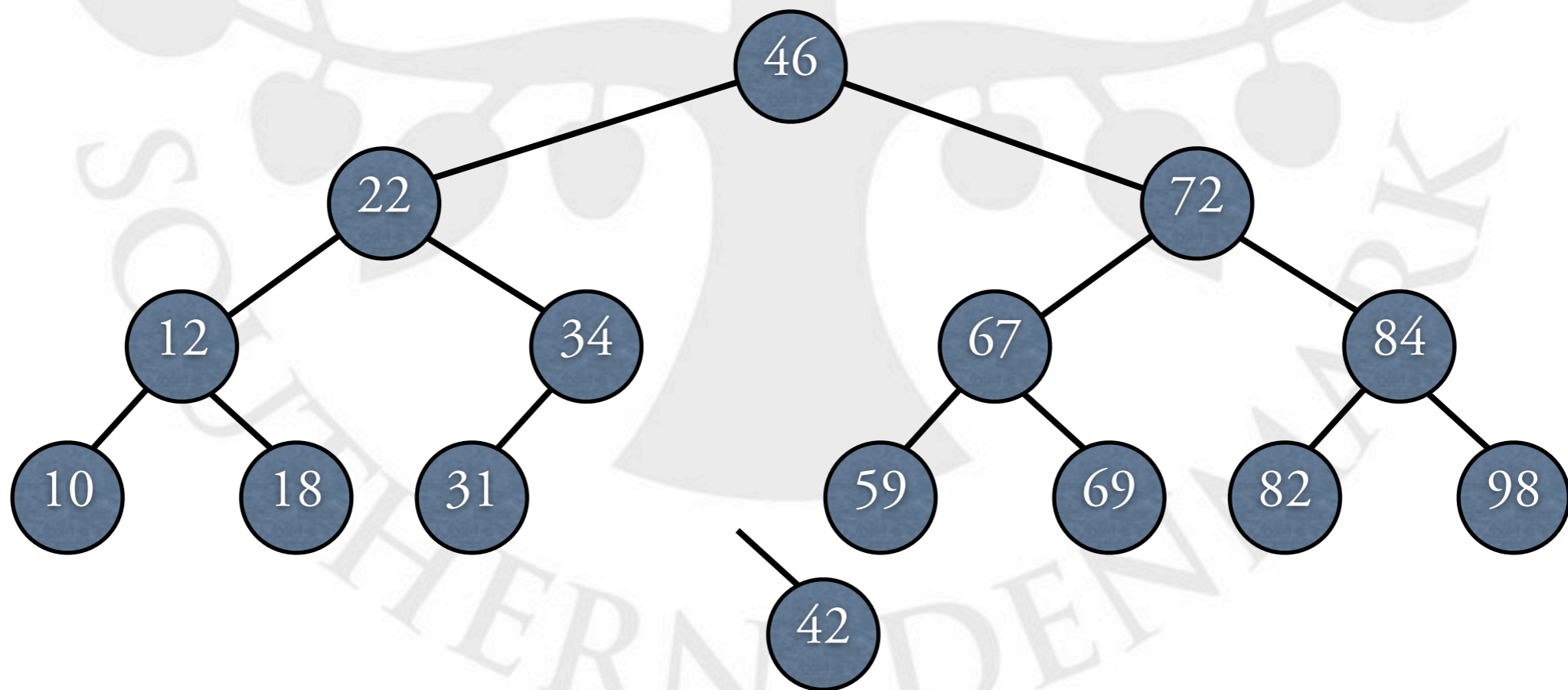
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med ét barn
 - Remove(40)



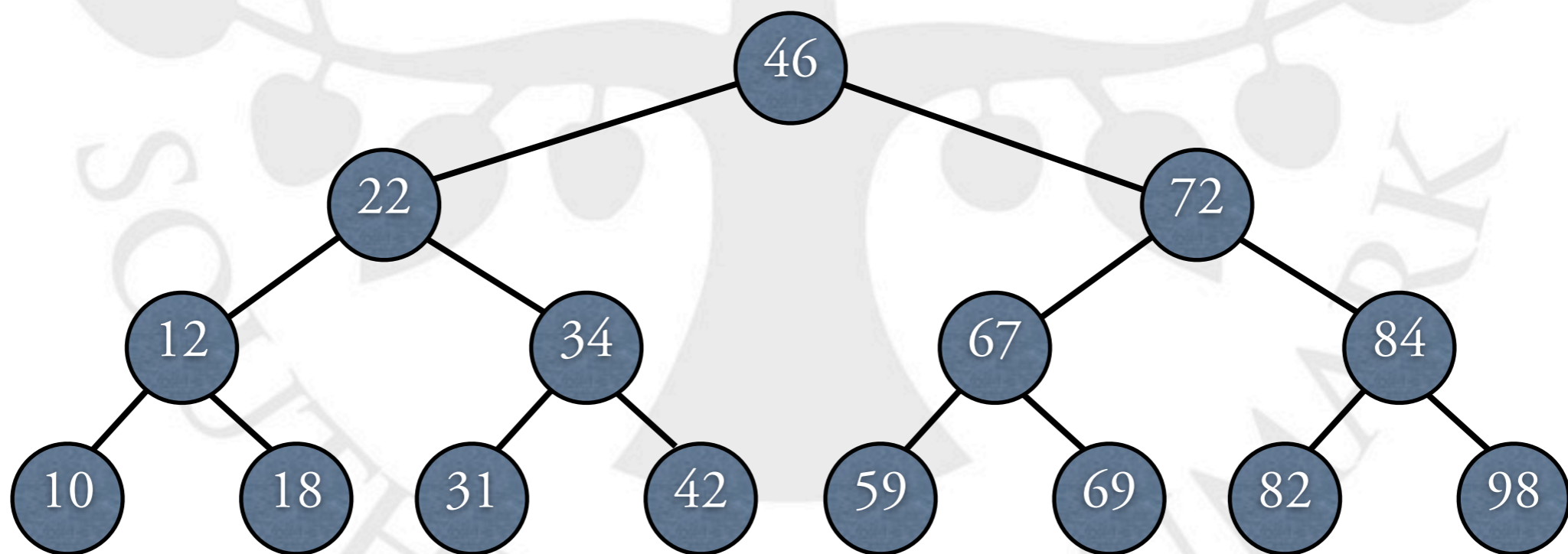
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med ét barn
 - Remove(40)

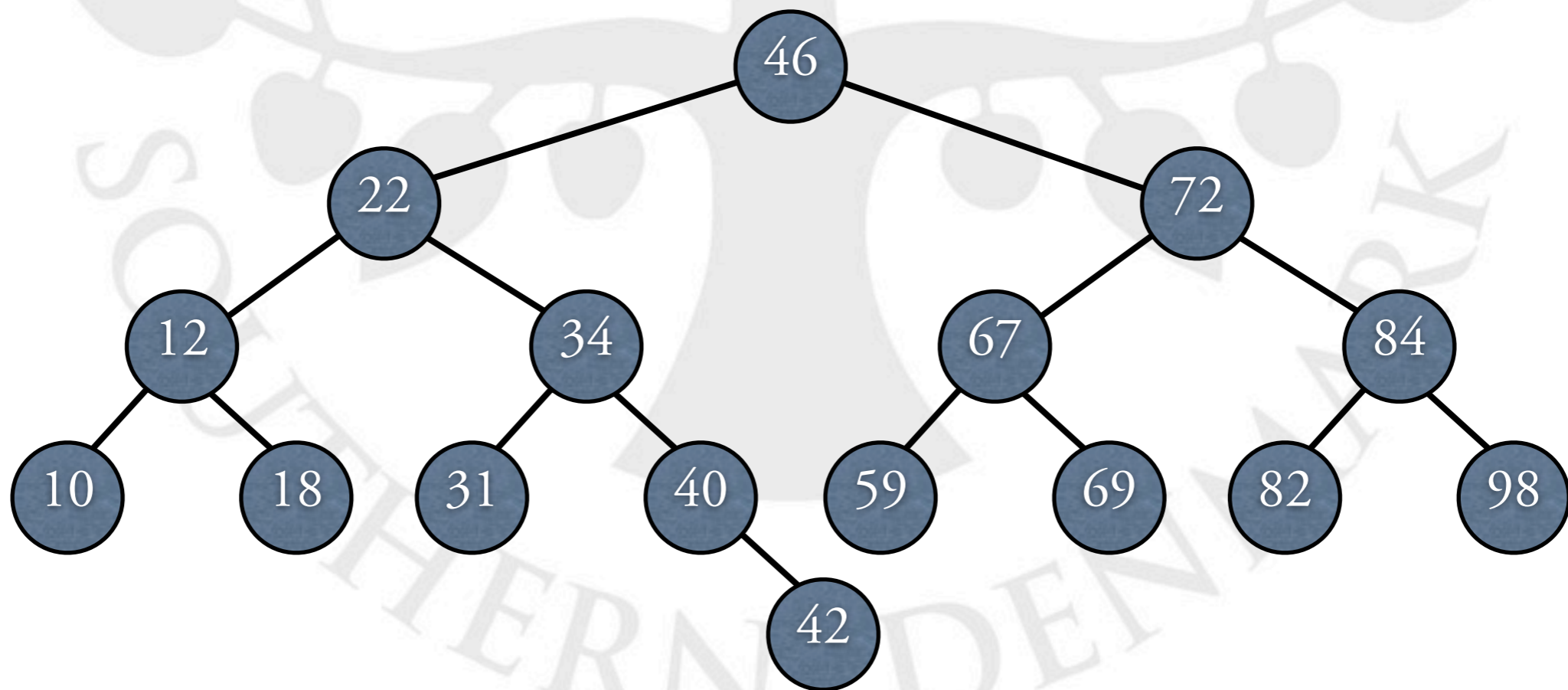


Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med ét barn
 - Remove(40)

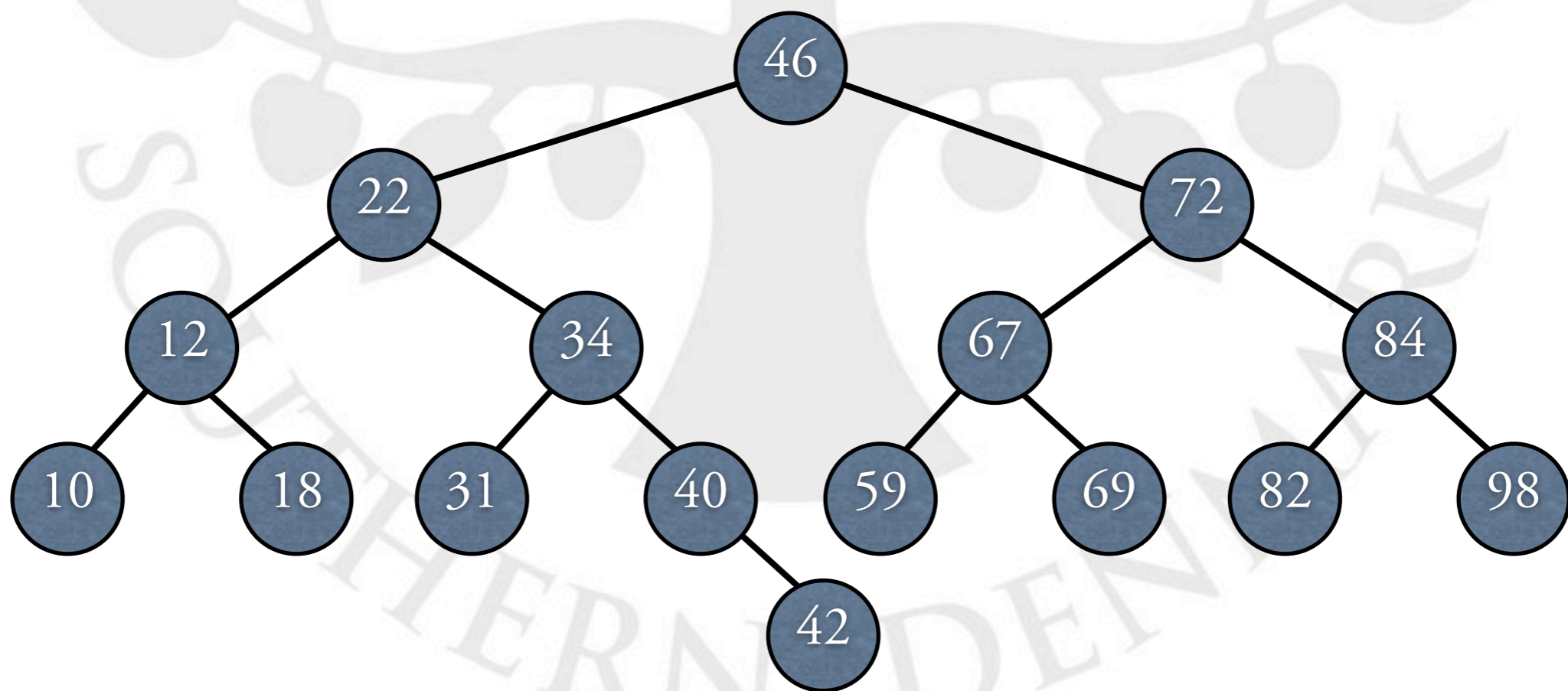


Binære søgetræer



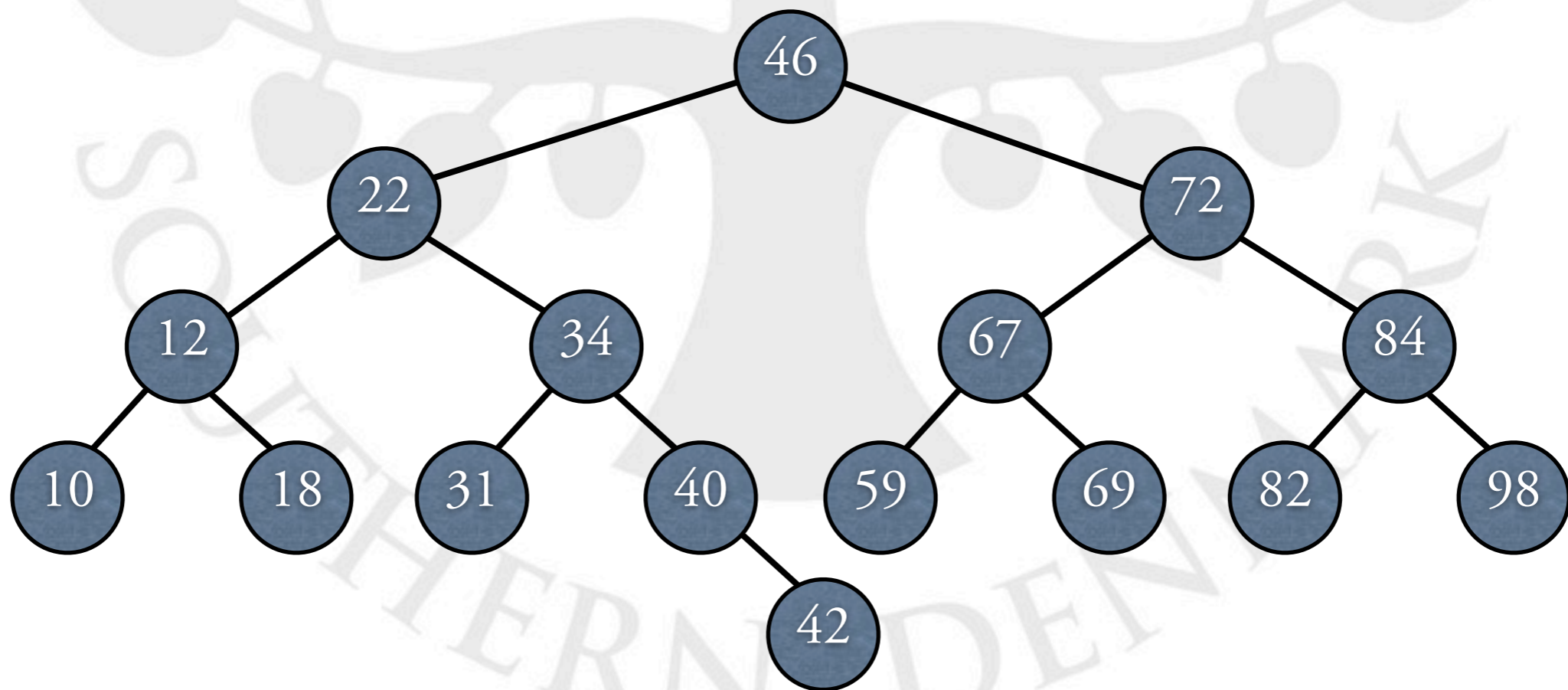
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde



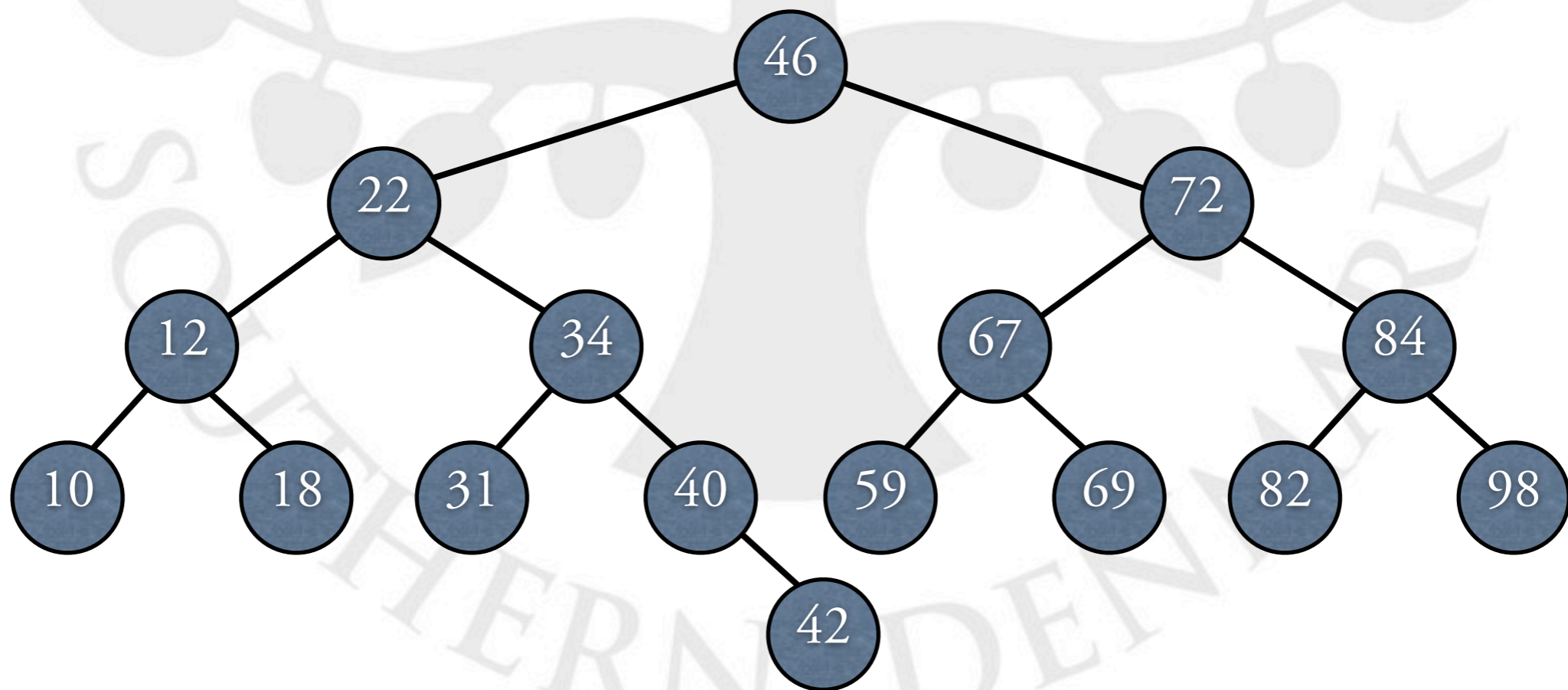
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med to børn



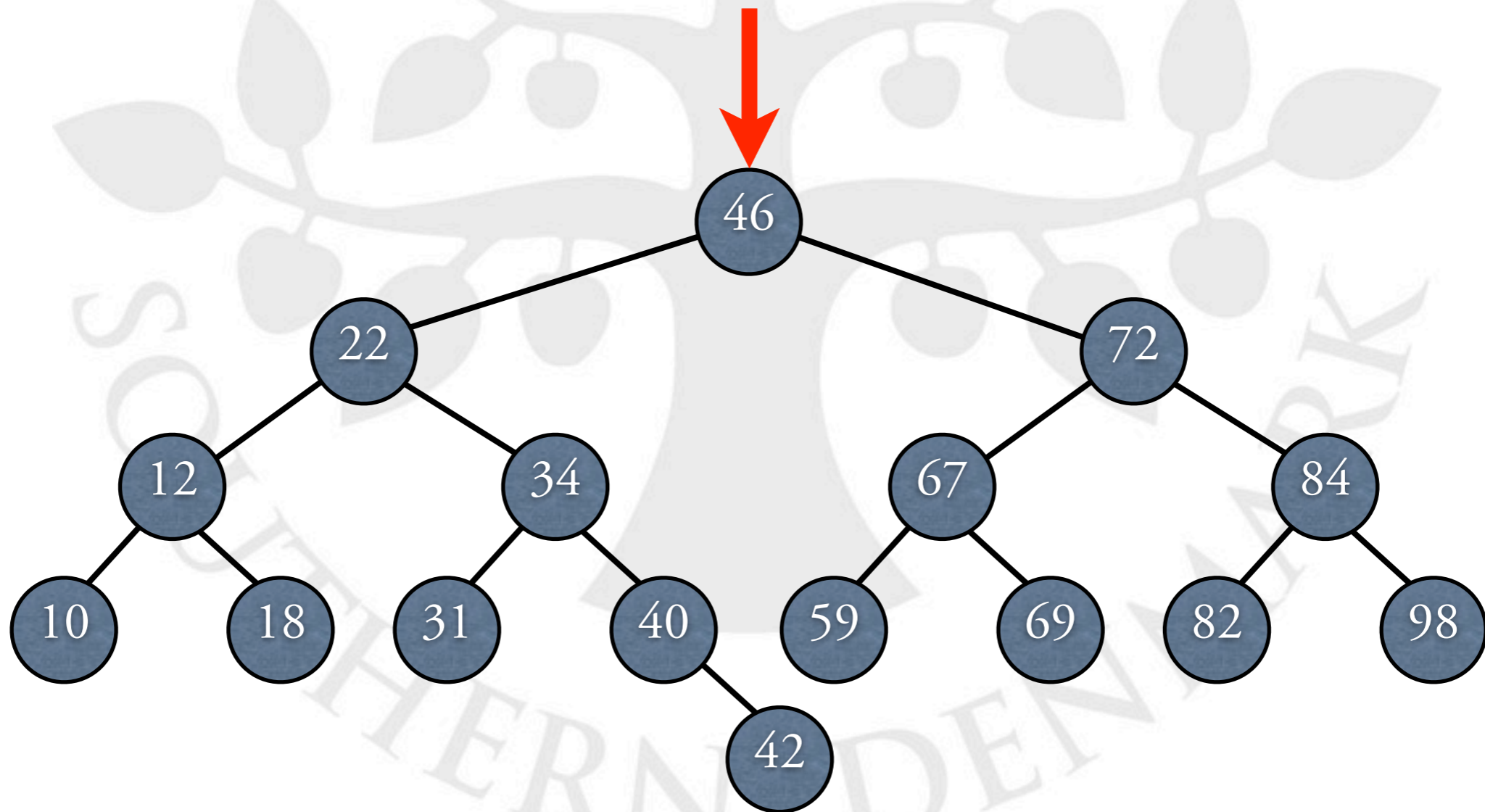
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med to børn
 - Remove(22)



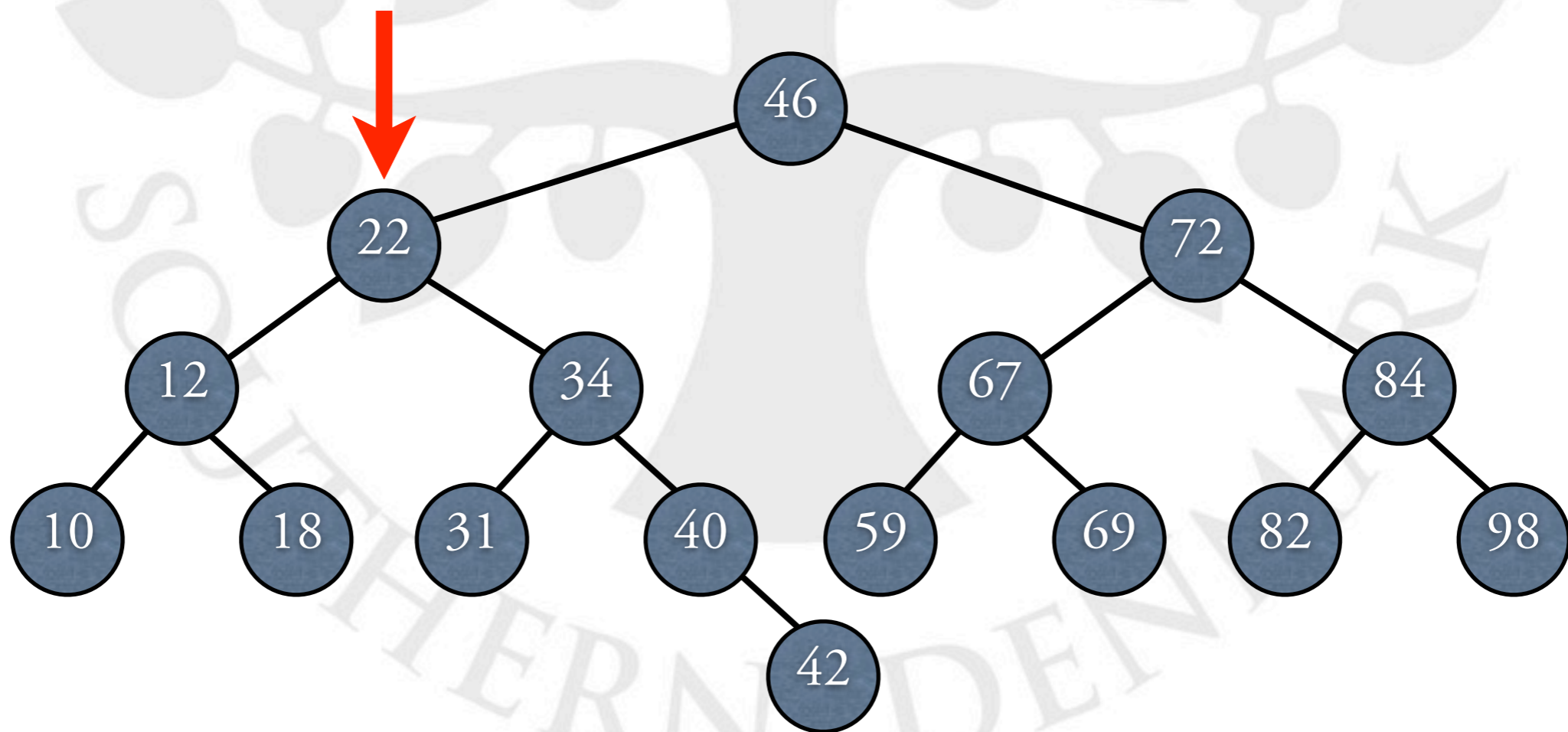
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med to børn
 - Remove(22)



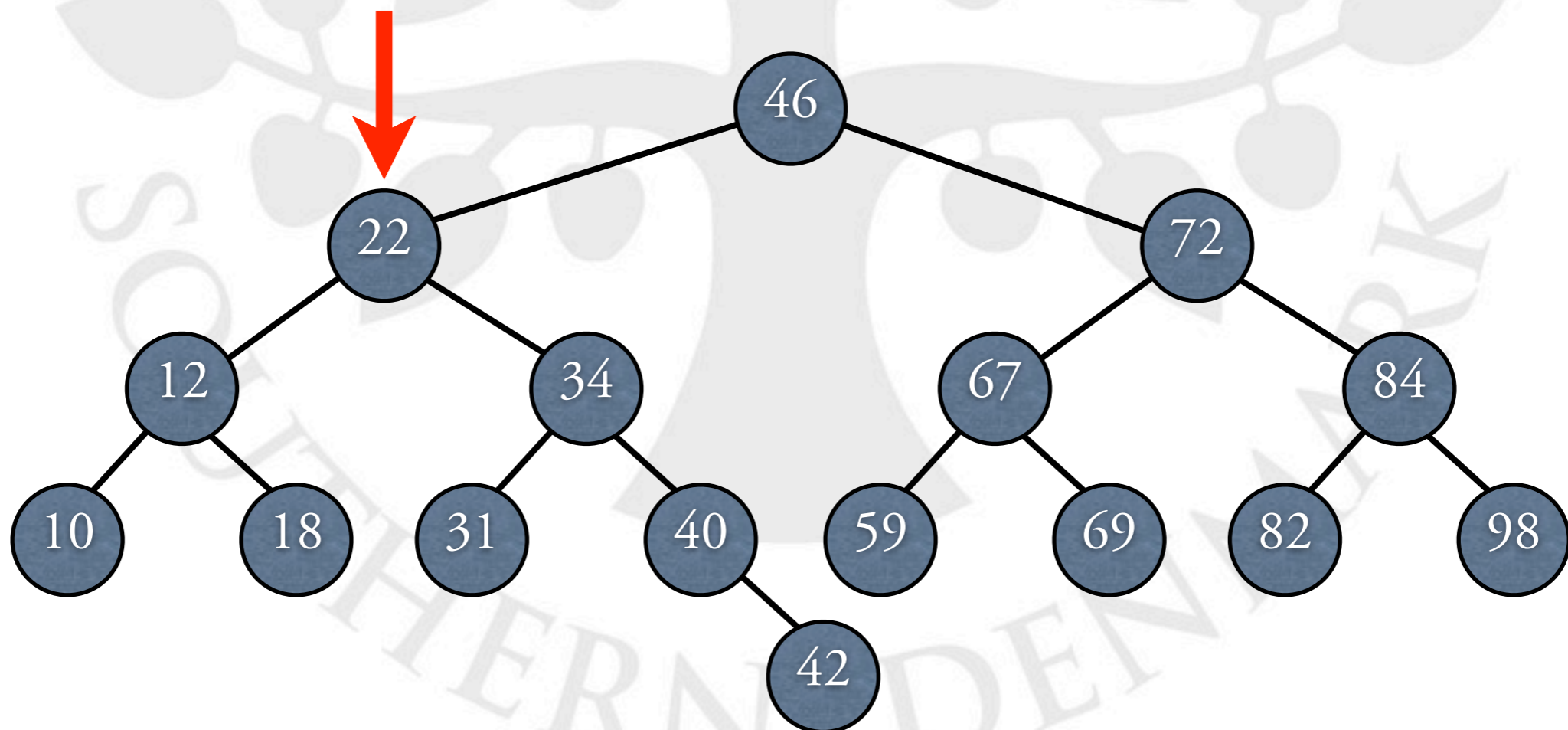
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med to børn
 - Remove(22)



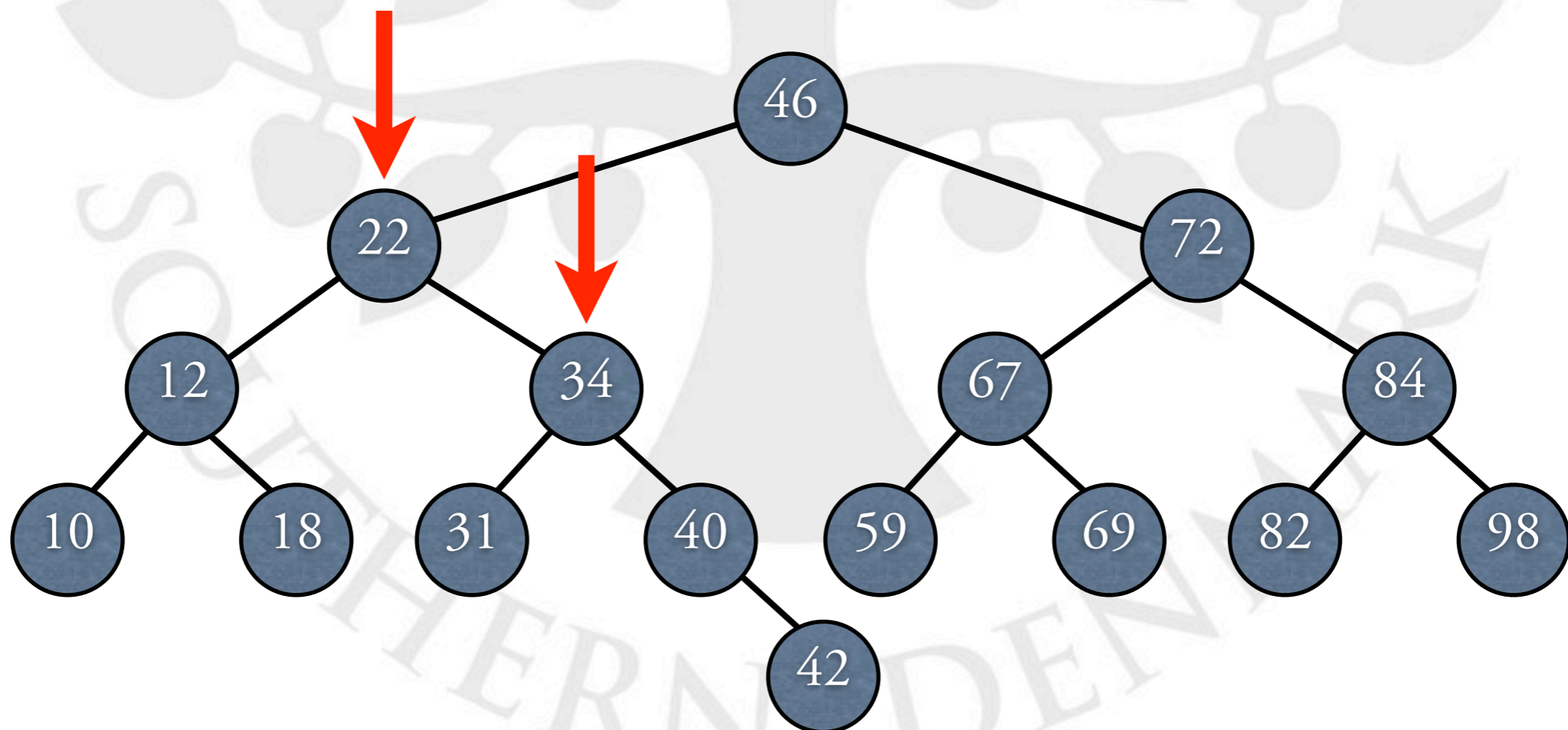
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med to børn
 - Remove(22)
 - Find next(22) (eller prev(22))



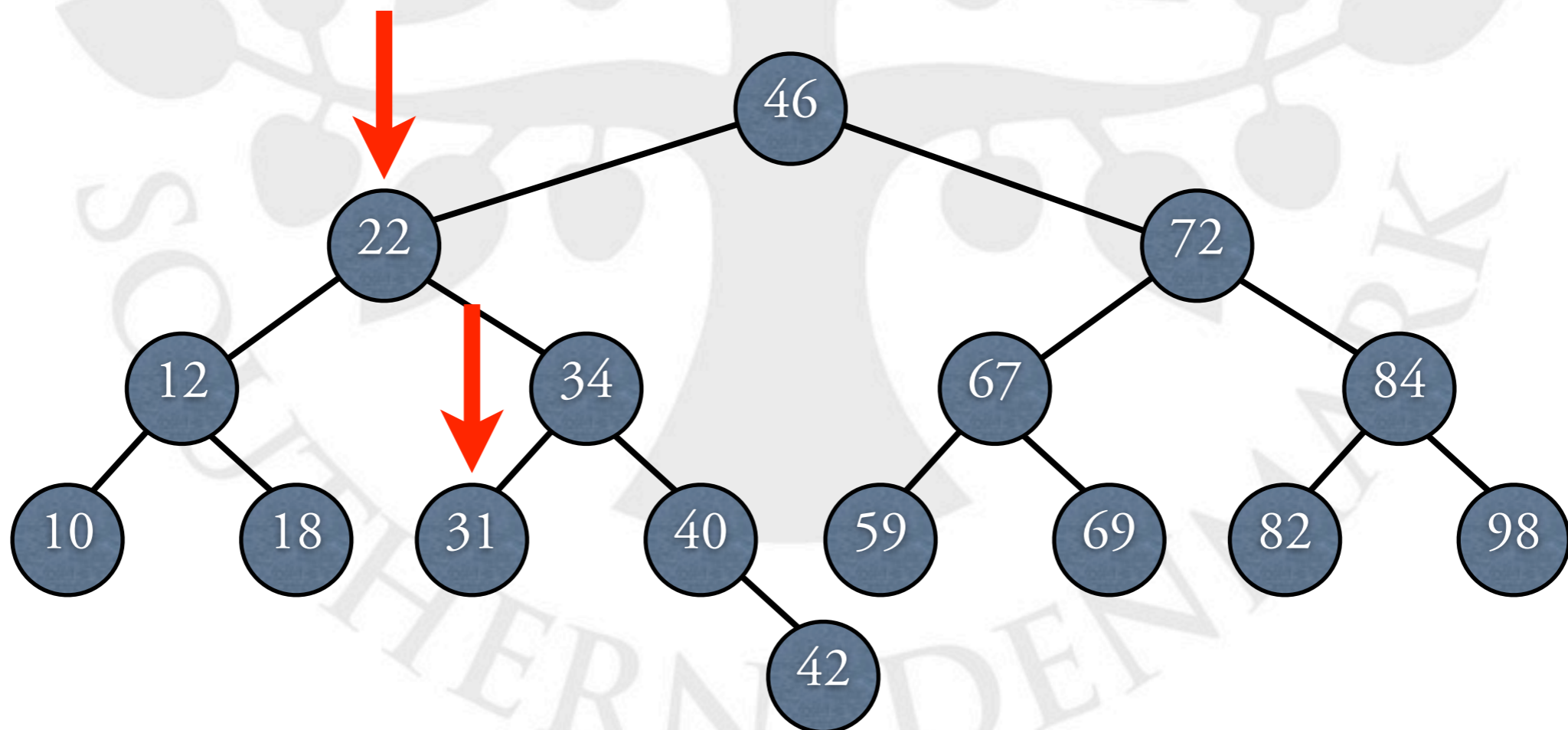
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med to børn
 - Remove(22)
 - Find next(22) (eller prev(22))



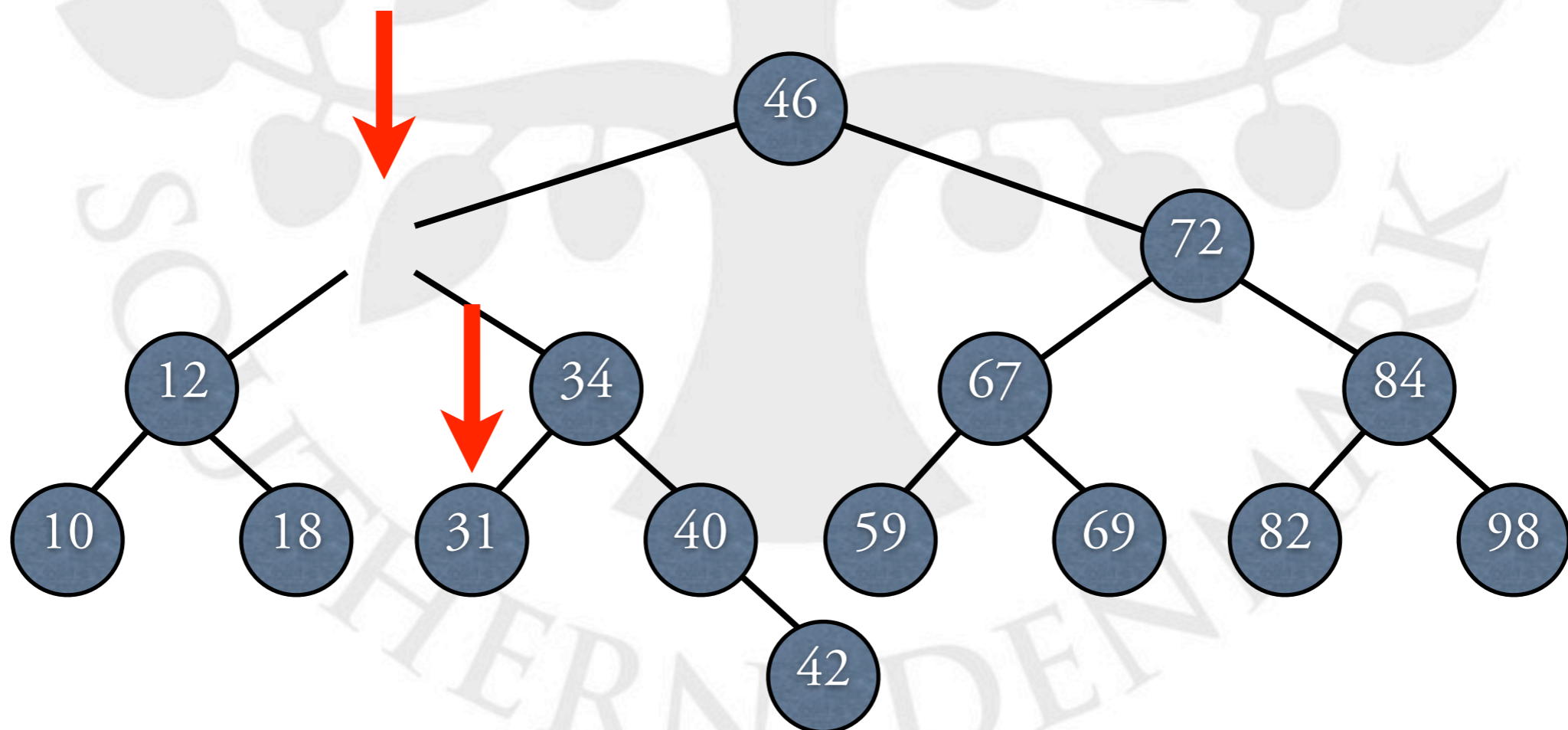
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med to børn
 - Remove(22)
 - Find next(22) (eller prev(22))



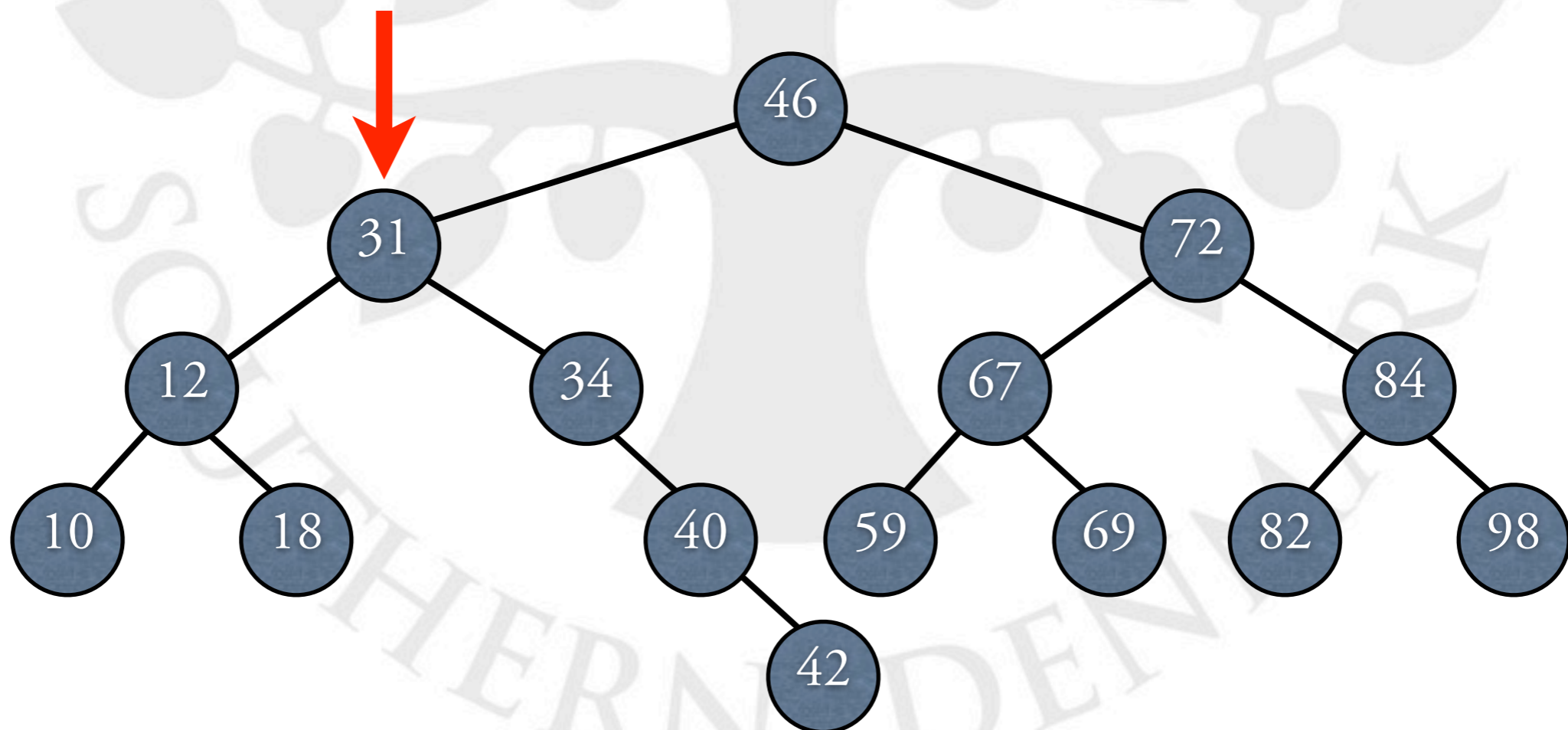
Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med to børn
 - Remove(22)
 - Find next(22) (eller prev(22))

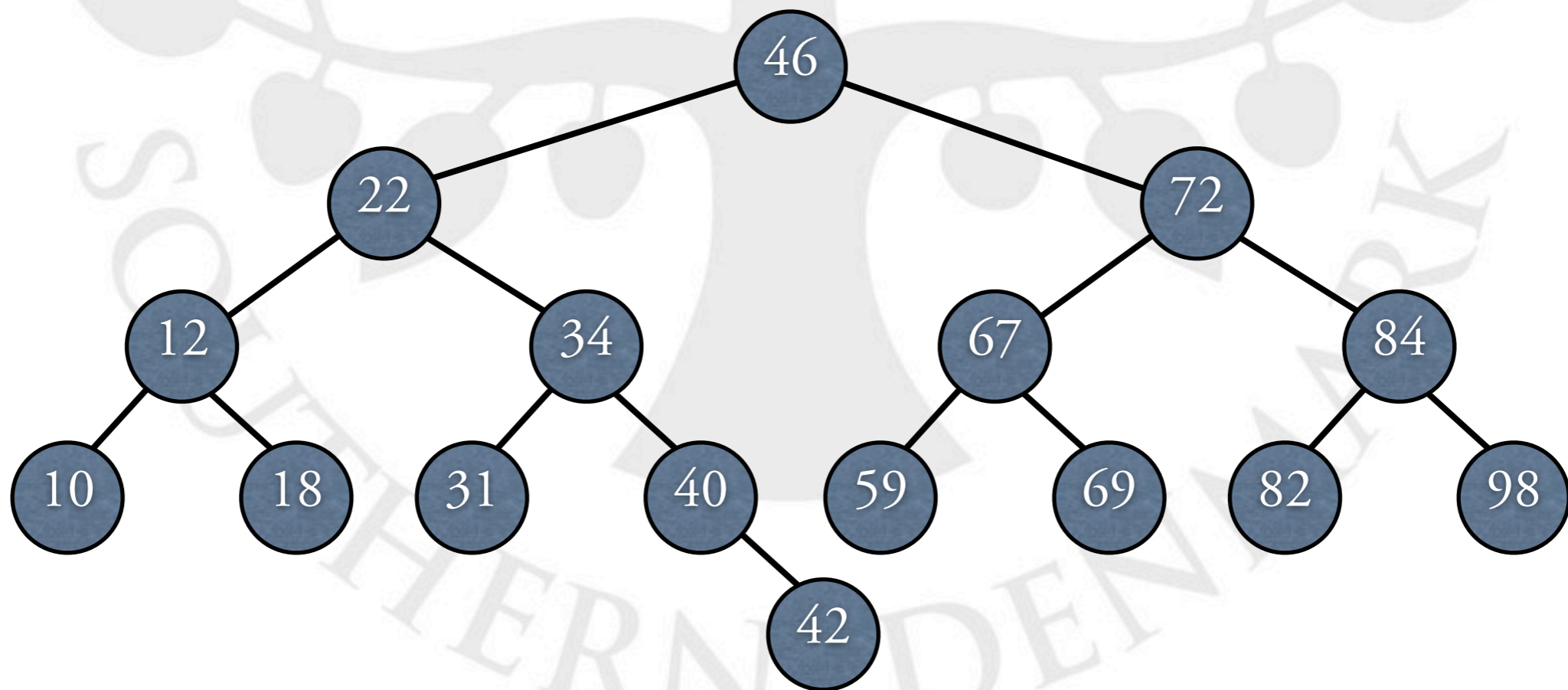


Binære søgetræer

- Hvad med sletninger?
 - Flere tilfælde
 - Sletning af knude med to børn
 - Remove(22)
 - Find next(22) (eller prev(22))

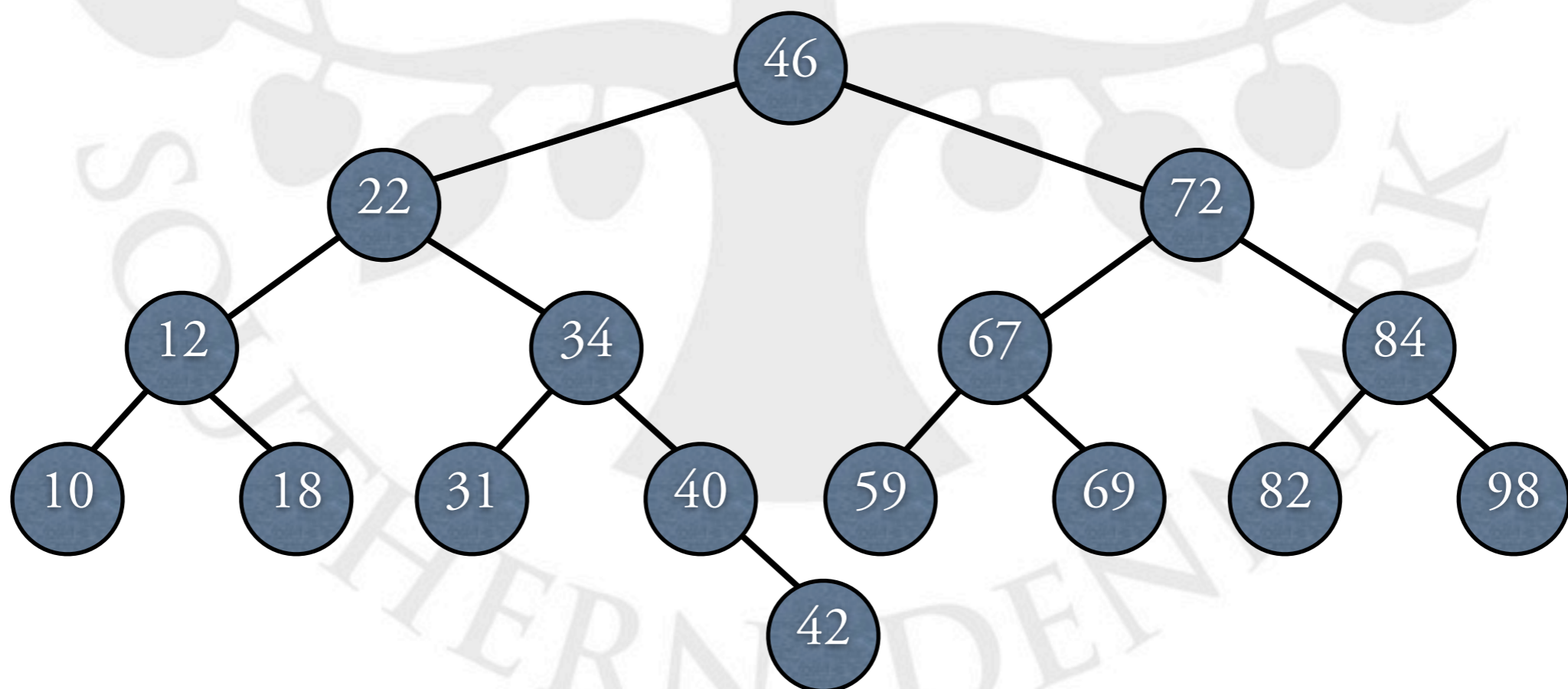


Binære søgetræer



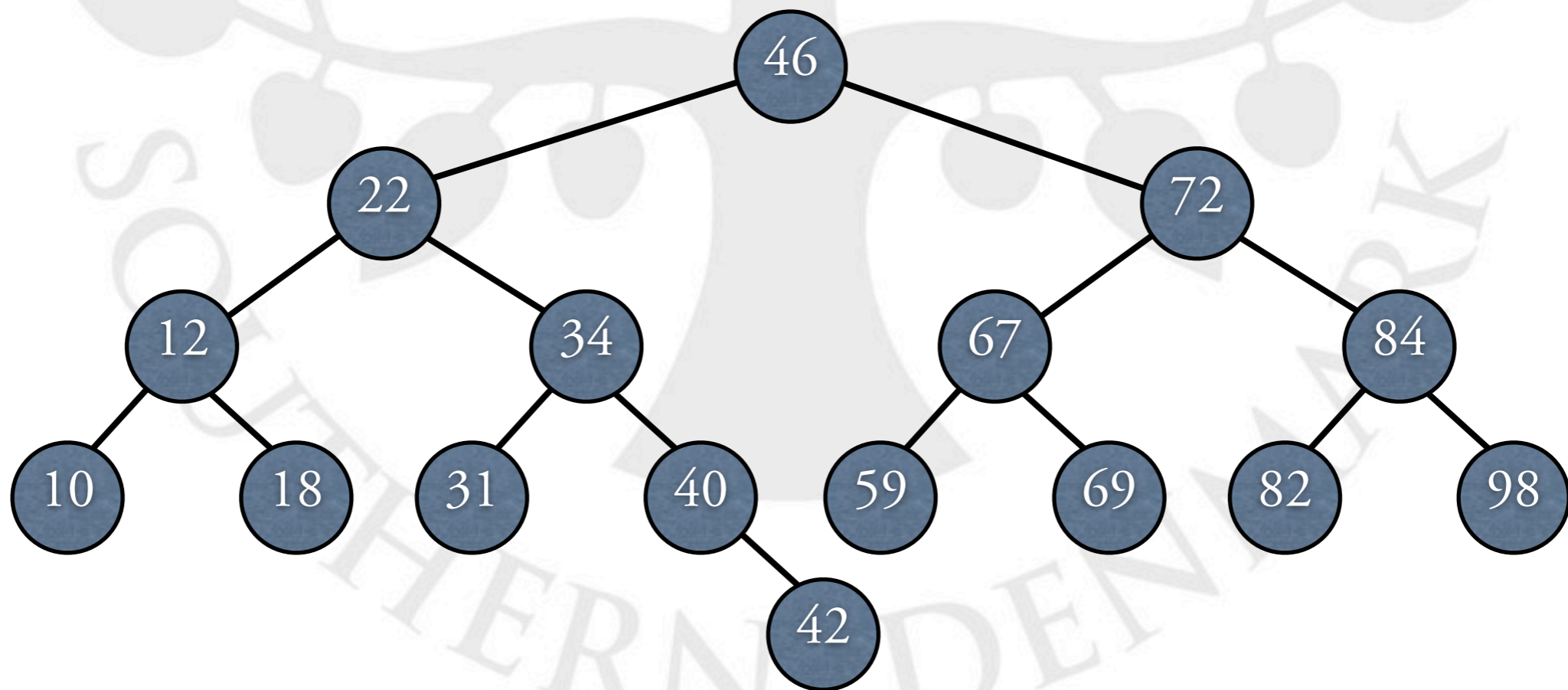
Binære søgetræer

- Hvor lang “tid” tager operationerne?



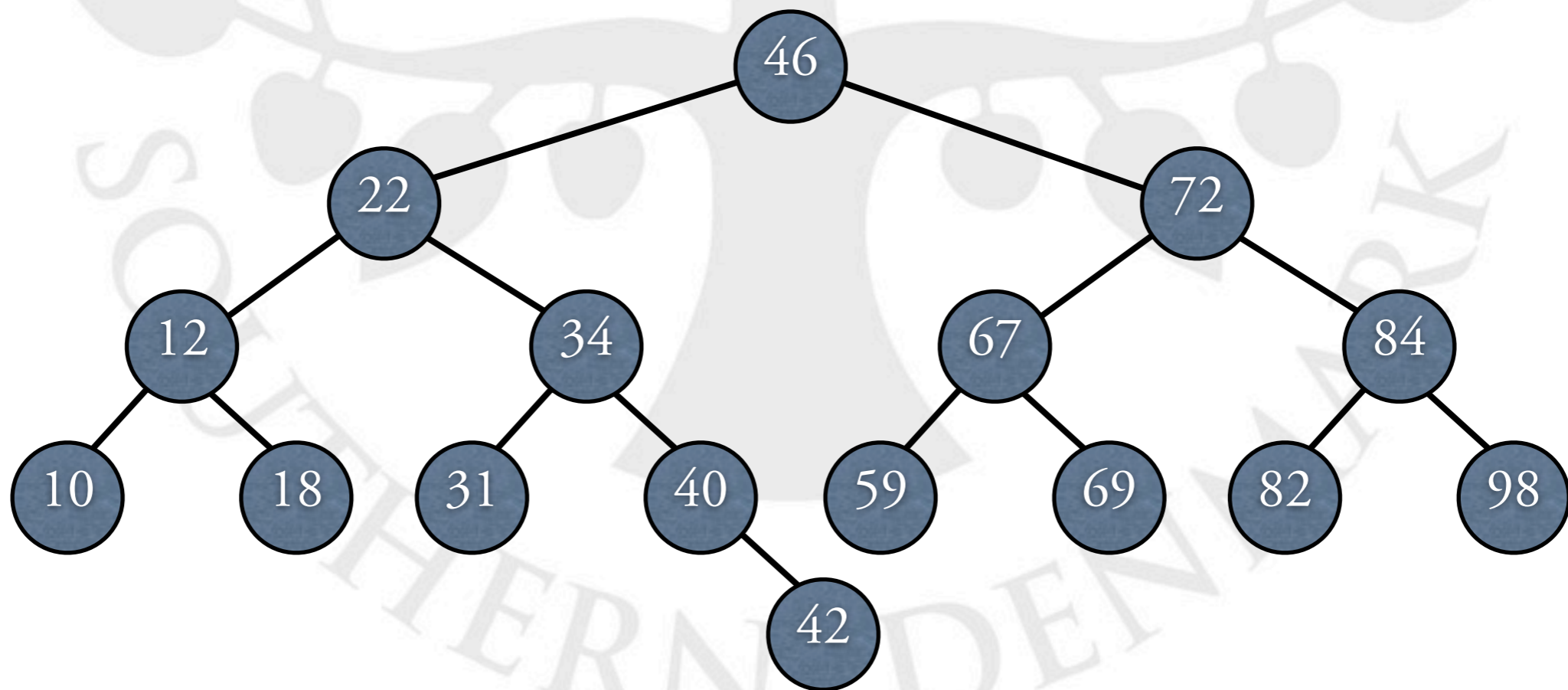
Binære søgetræer

- Hvor lang “tid” tager operationerne?
 - Det der tager tid er Search



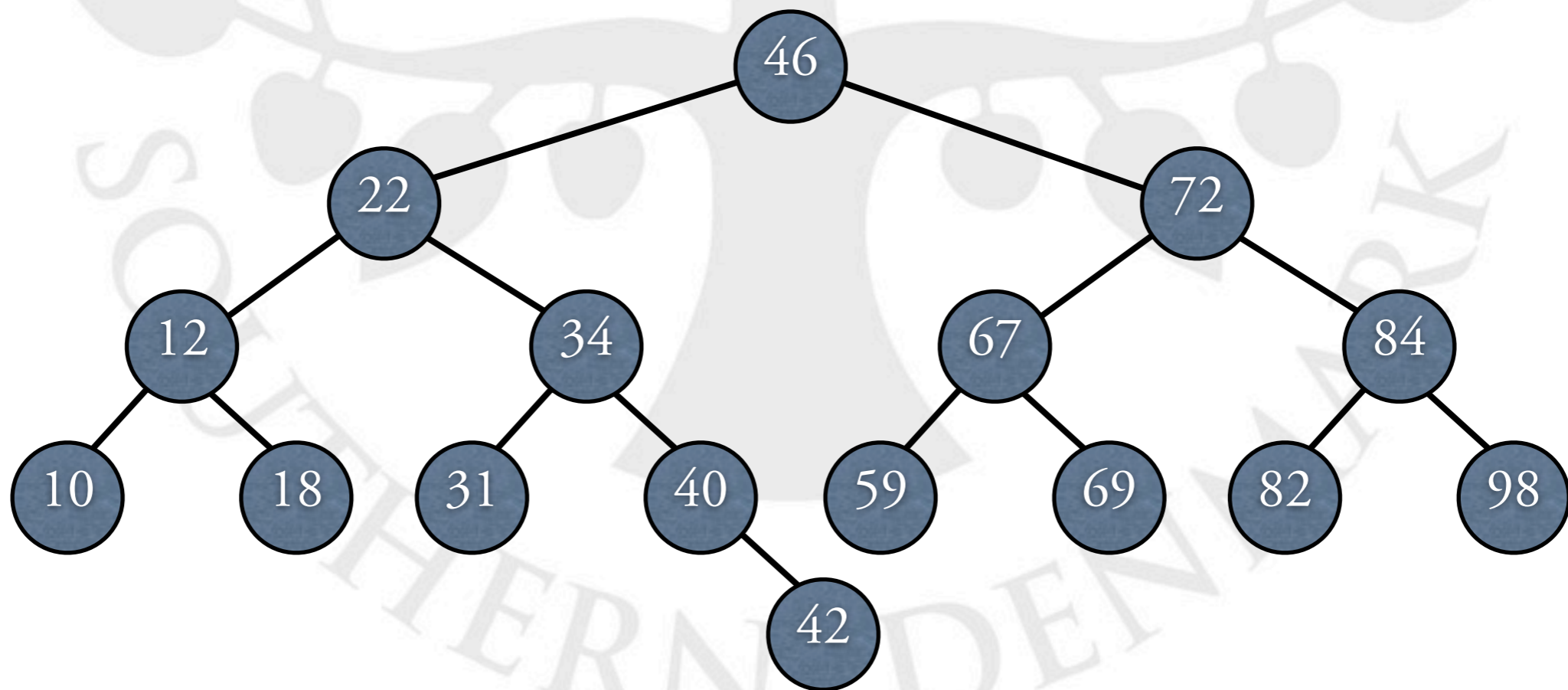
Binære søgetræer

- Hvor lang “tid” tager operationerne?
 - Det der tager tid er Search
 - I værste tilfælde skal Search kigge på et antal knuder der svarer til højden af træet



Binære søgetræer

- Hvor lang “tid” tager operationerne?
 - Det der tager tid er Search
 - I værste tilfælde skal Search kigge på et antal knuder der svarer til højden af træet
 - Sørg for at højden er lav (DM507)





1. delprojekt

Sudoku Solver

- Abstract Data Type for Sudoku
- løser, der bruger ADT og REKURSION
- skabeloner på kursets hjemmeside
- aflevering senest 2. december kl. 12
- alt andet som i DM502



Radix-træer

- Sortering af binære tal vha. radix-træer
 - Indlæsning af binære tal fra en fil
 - Opbygning af et radix-træ over tallene
 - Pre-order gennemløb af træet, således at tallene udskrives sorteret i leksikografisk orden



Radix-træer over binære tal



Radix-træer over binære tal

- Et binært træ der indeholder de binære tal



Radix-træer over binære tal

- Et binært træ der indeholder de binære tal
- Tallene gemmes IKKE i knuderne



Radix-træer over binære tal

- Et binært træ der indeholder de binære tal
- Tallene gemmes IKKE i knuderne
- Tallene gemmes som stier fra roden til knuderne



Radix-træer over binære tal

- Et binært træ der indeholder de binære tal
- Tallene gemmes IKKE i knuderne
- Tallene gemmes som stier fra roden til knuderne
- To typer af knuder



Radix-træer over binære tal

- Et binært træ der indeholder de binære tal
- Tallene gemmes IKKE i knuderne
- Tallene gemmes som stier fra roden til knuderne
- To typer af knuder
 - Sorte knuder: Stien fra roden til knuden repræsenterer et binært tal fra input-filen

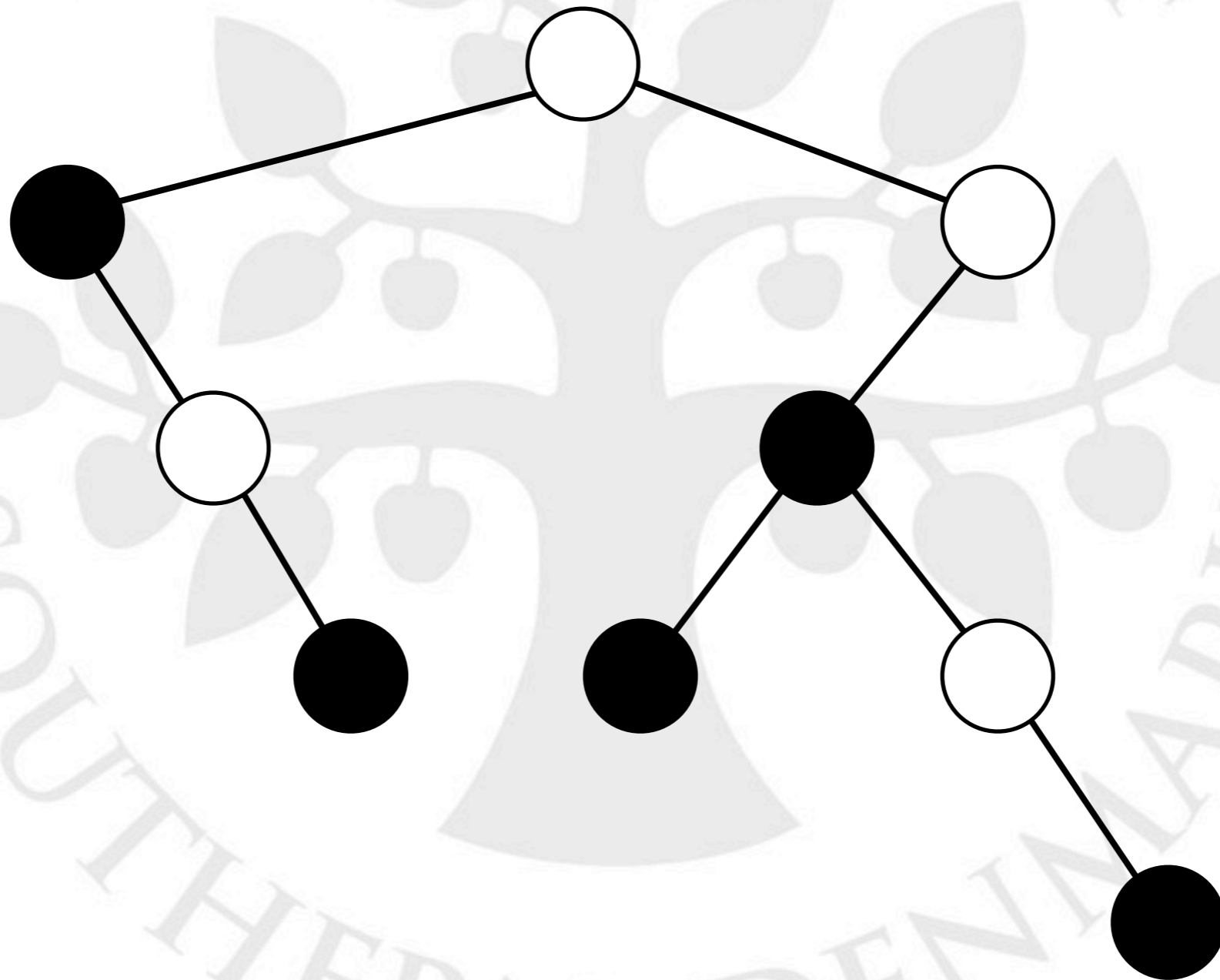


Radix-træer over binære tal

- Et binært træ der indeholder de binære tal
- Tallene gemmes IKKE i knuderne
- Tallene gemmes som stier fra roden til knuderne
- To typer af knuder
 - Sorte knuder: Stien fra roden til knuden repræsenterer et binært tal fra input-filen
 - Hvide knuder: Stien fra roden til knuden repræsenterer ikke et binært tal fra input-filen

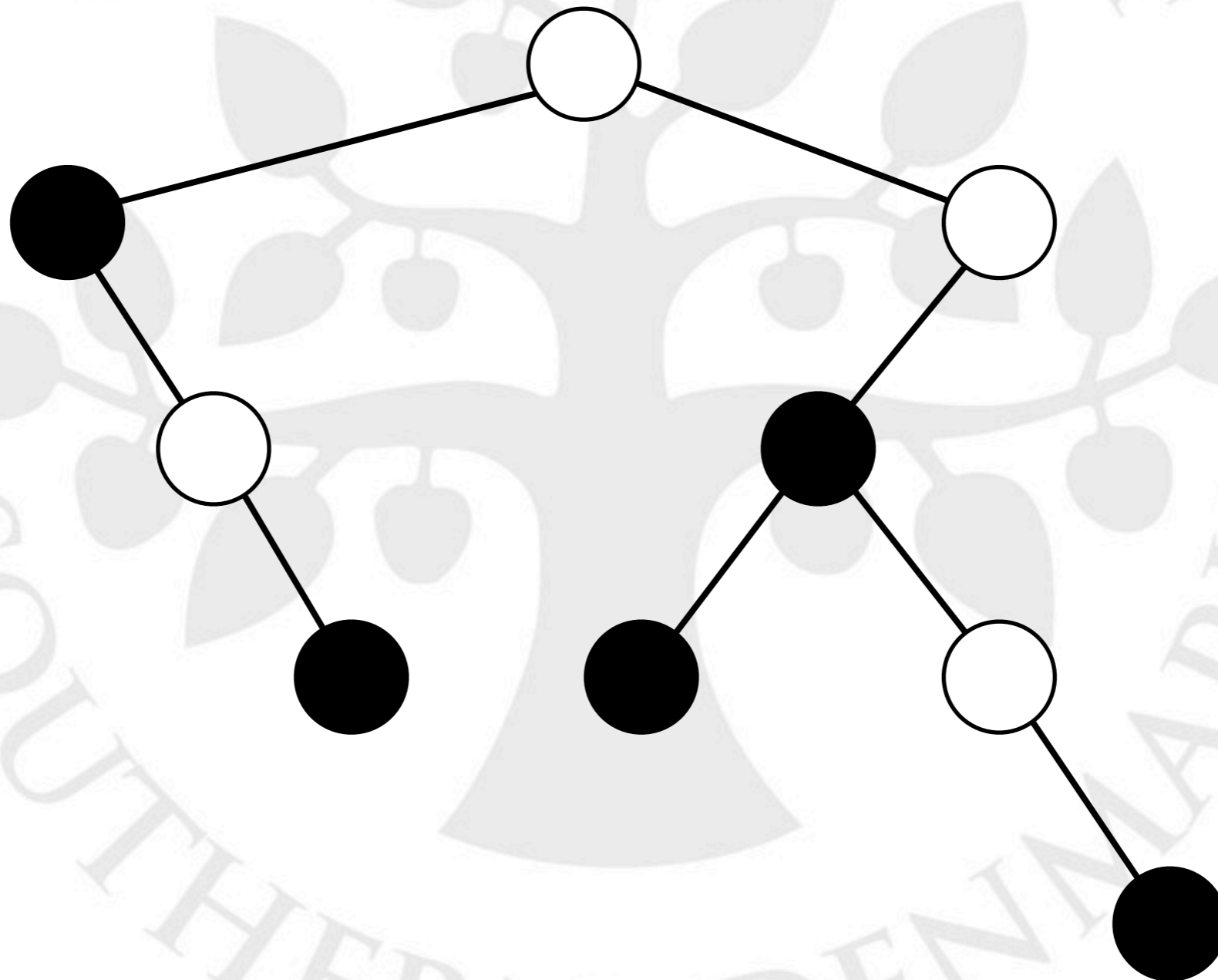


Radix-træer over binære tal



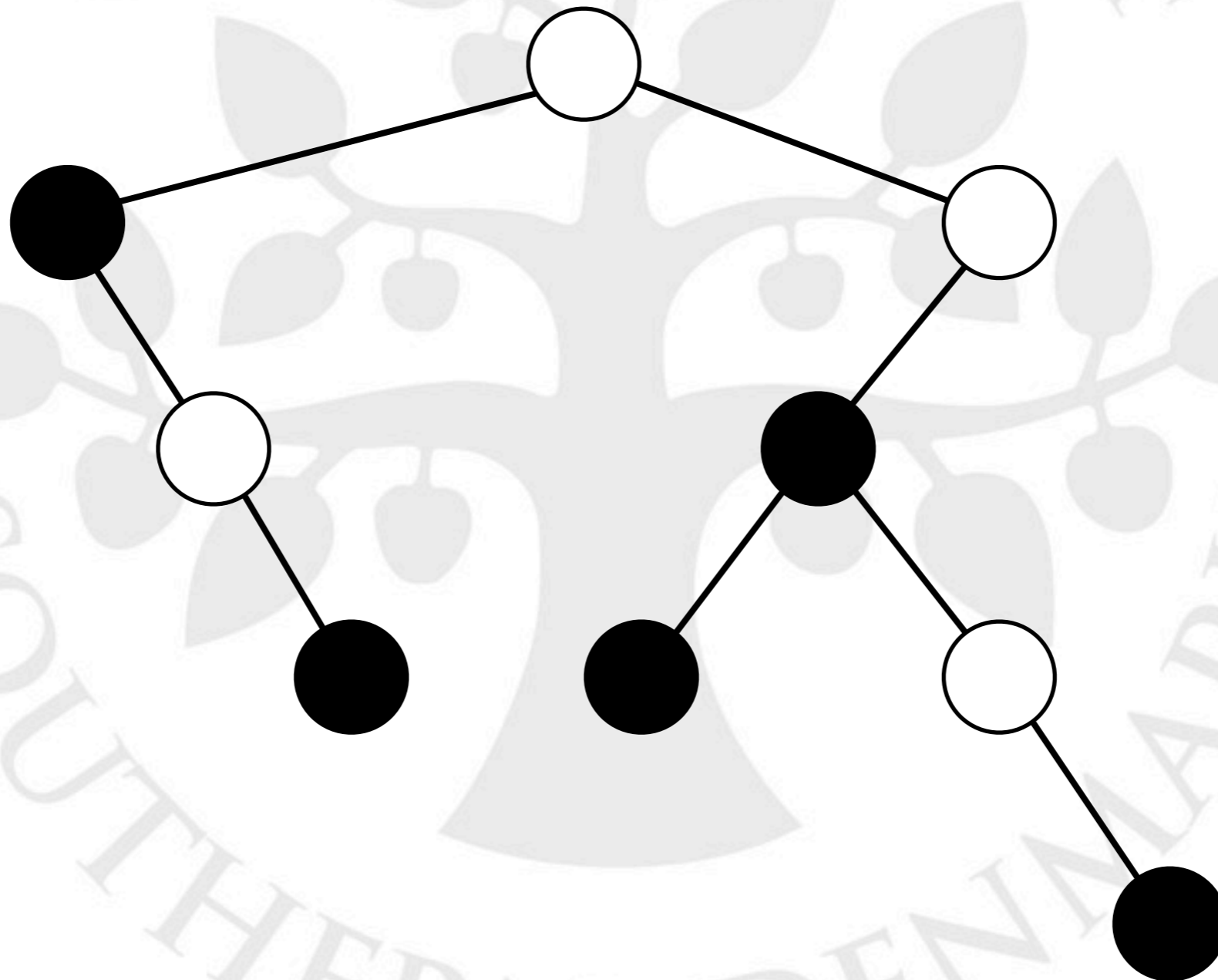
Radix-træer over binære tal

- Eksempel



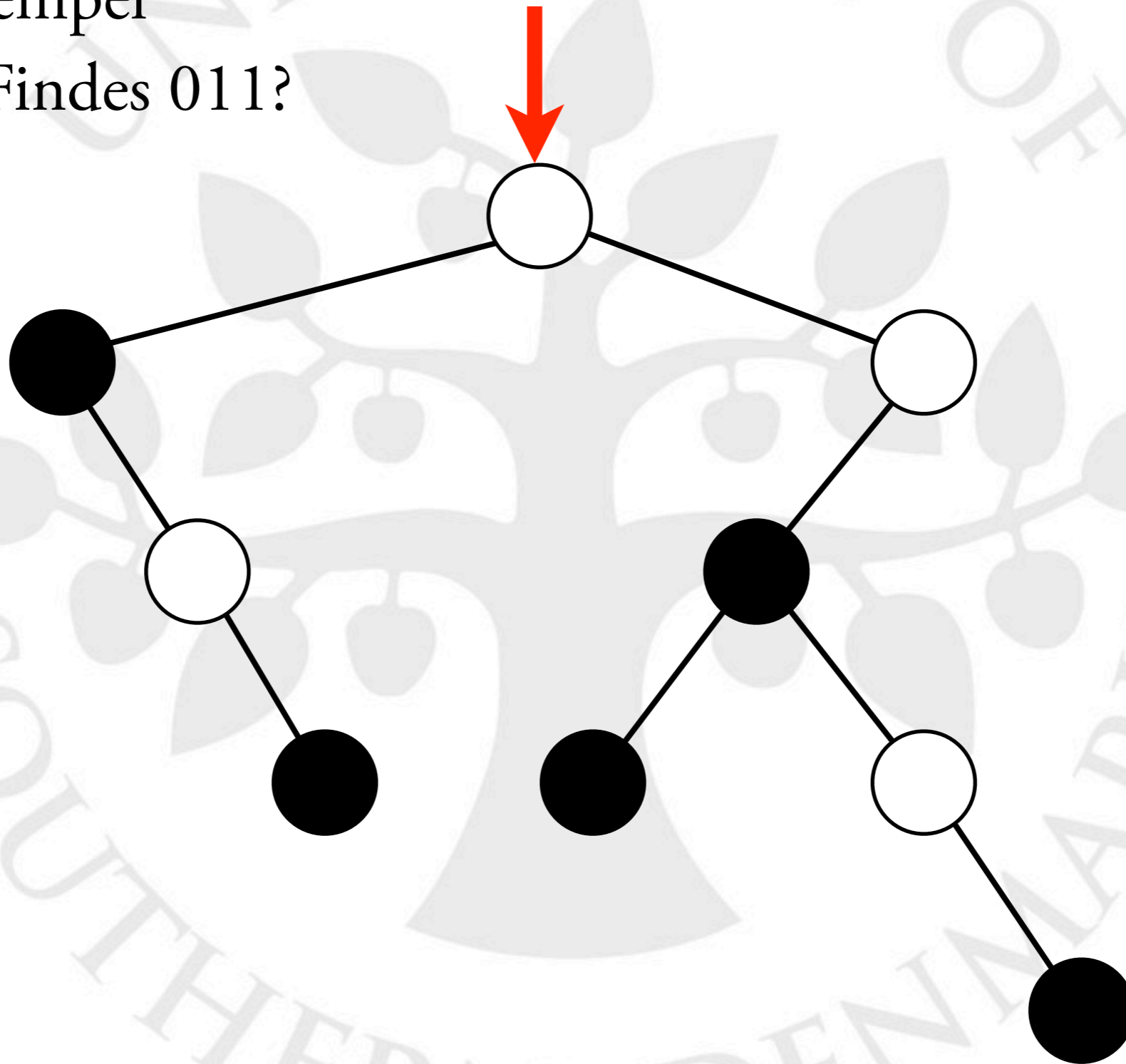
Radix-træer over binære tal

- Eksempel
 - Findes 011?



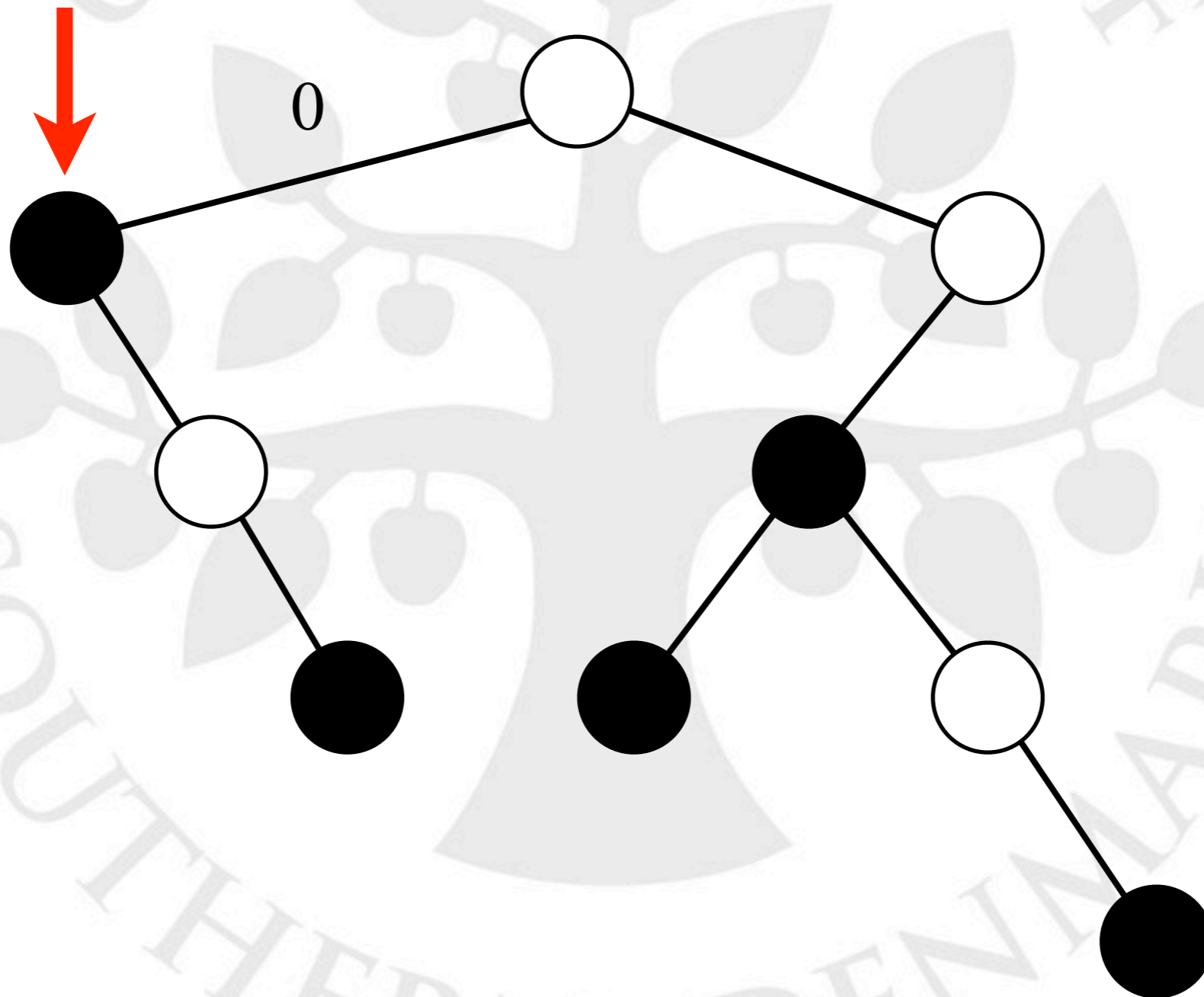
Radix-træer over binære tal

- Eksempel
 - Findes 011?



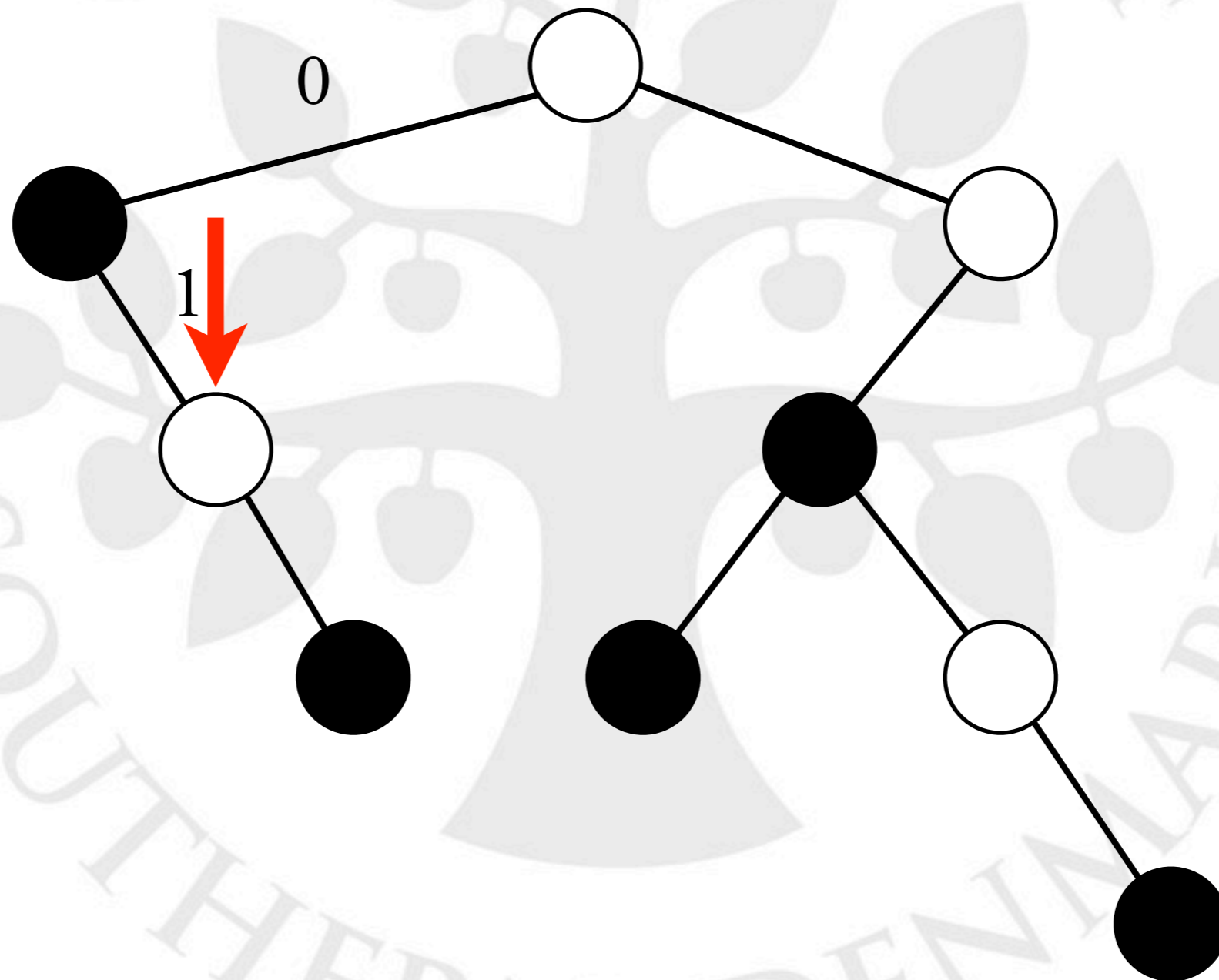
Radix-træer over binære tal

- Eksempel
 - Findes 011?



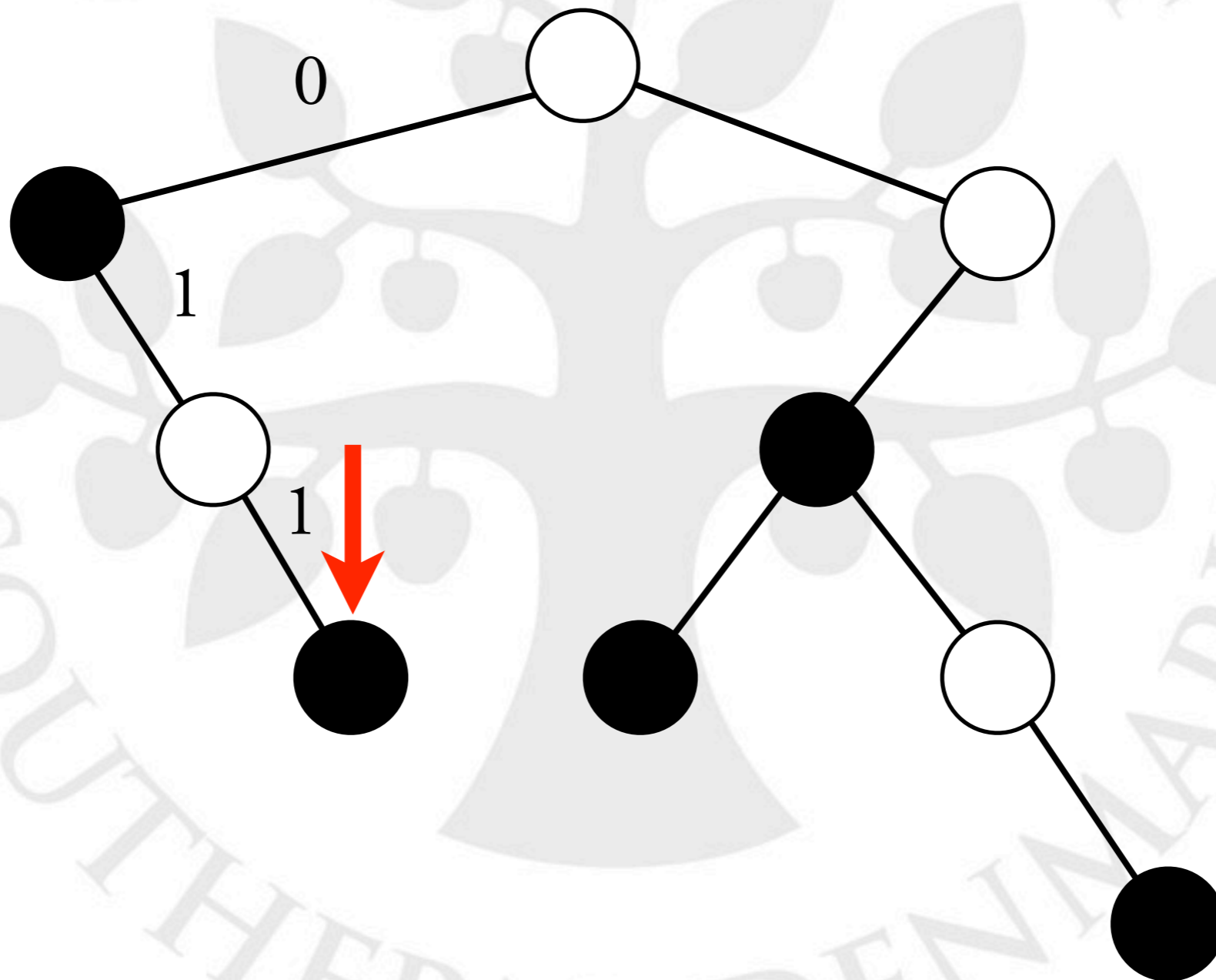
Radix-træer over binære tal

- Eksempel
 - Findes 011?



Radix-træer over binære tal

- Eksempel
 - Findes 011?

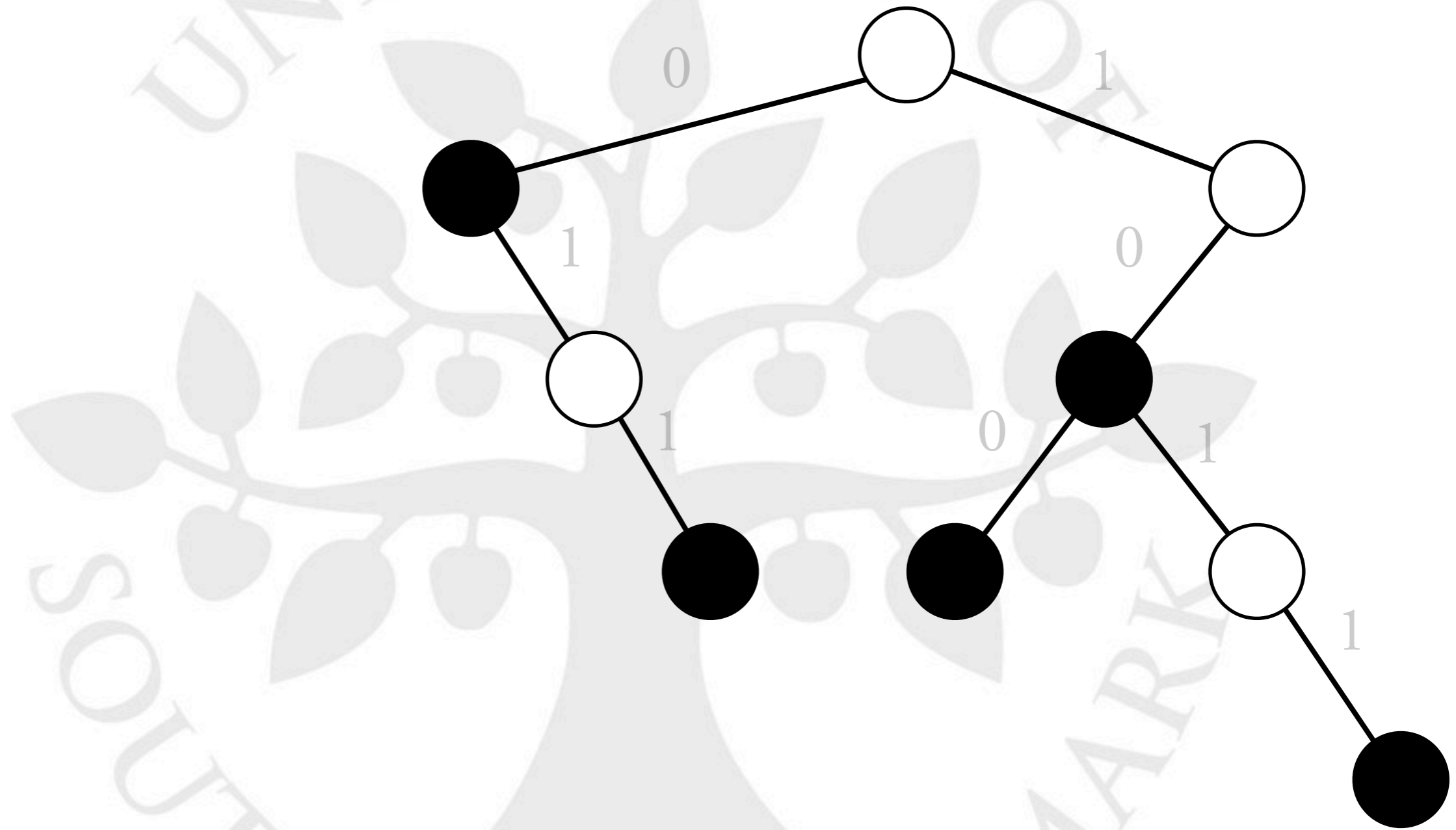


Radix-træer over binære tal

- Søgning efter $a = a_0 a_1 \dots a_n$ (fx 011)
 - Start i roden
 - Kig på det første ciffer (0)
 - Hvis 0 gå til venstre barn
 - Hvis 1 gå til højre barn
 - Fortsæt således for hvert ciffer
 - Hvis den knude man ender med er sort findes tallet
 - Hvis den knude man ender med er hvid, eller hvis man skal følge en kant til en knude der ikke findes findes tallet ikke

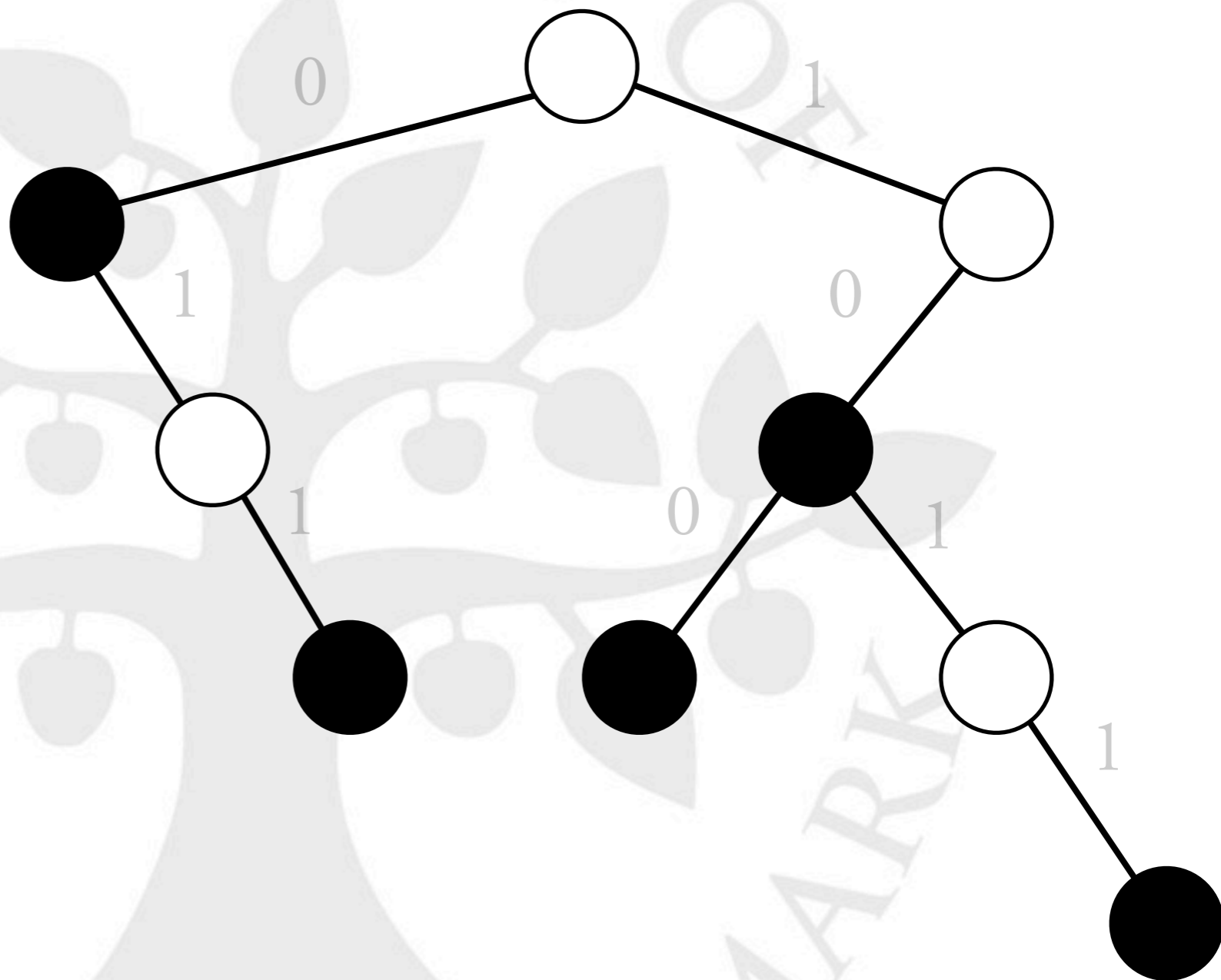


Radix-træer over binære tal



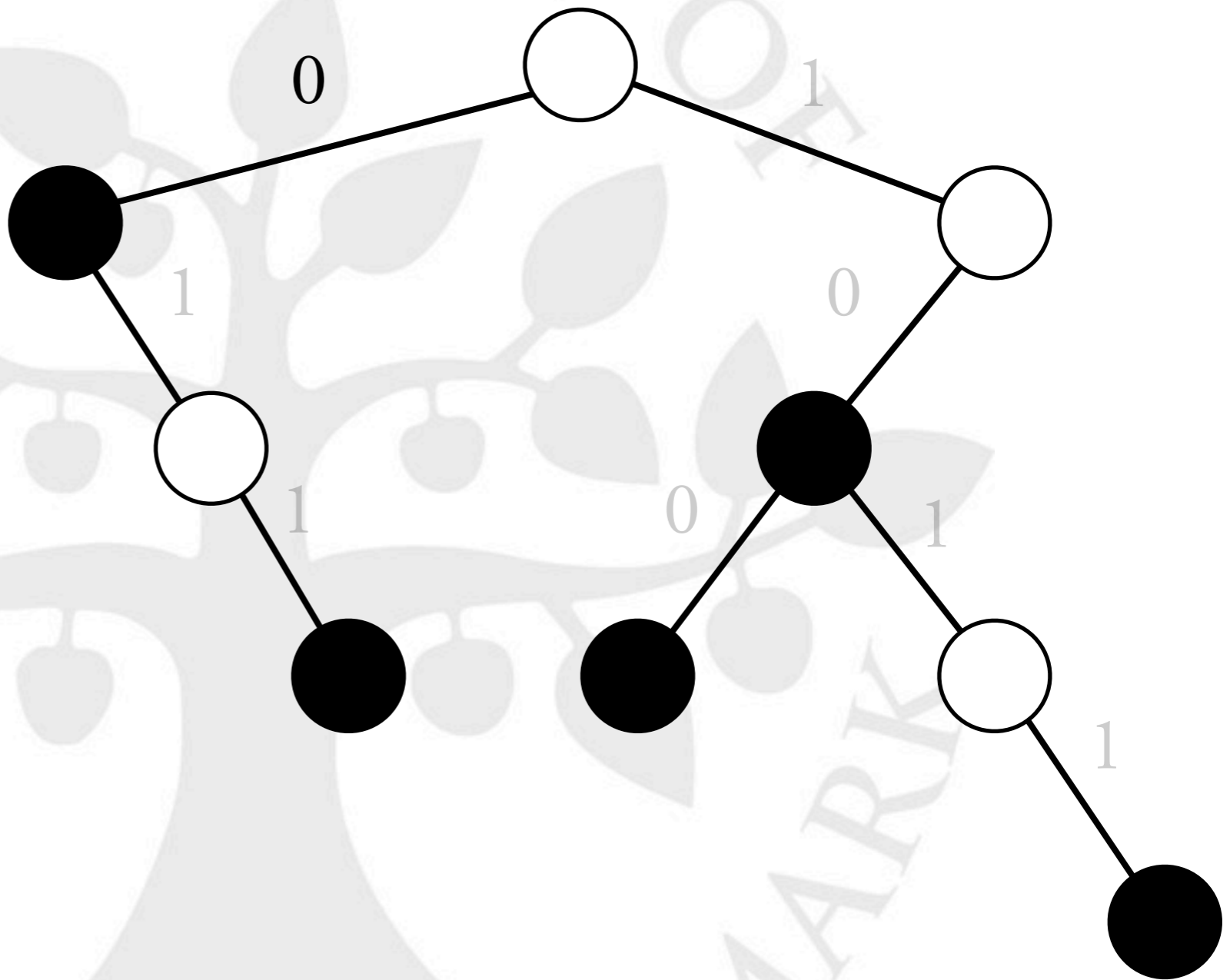
Radix-træer over binære tal

- Tal i træet?



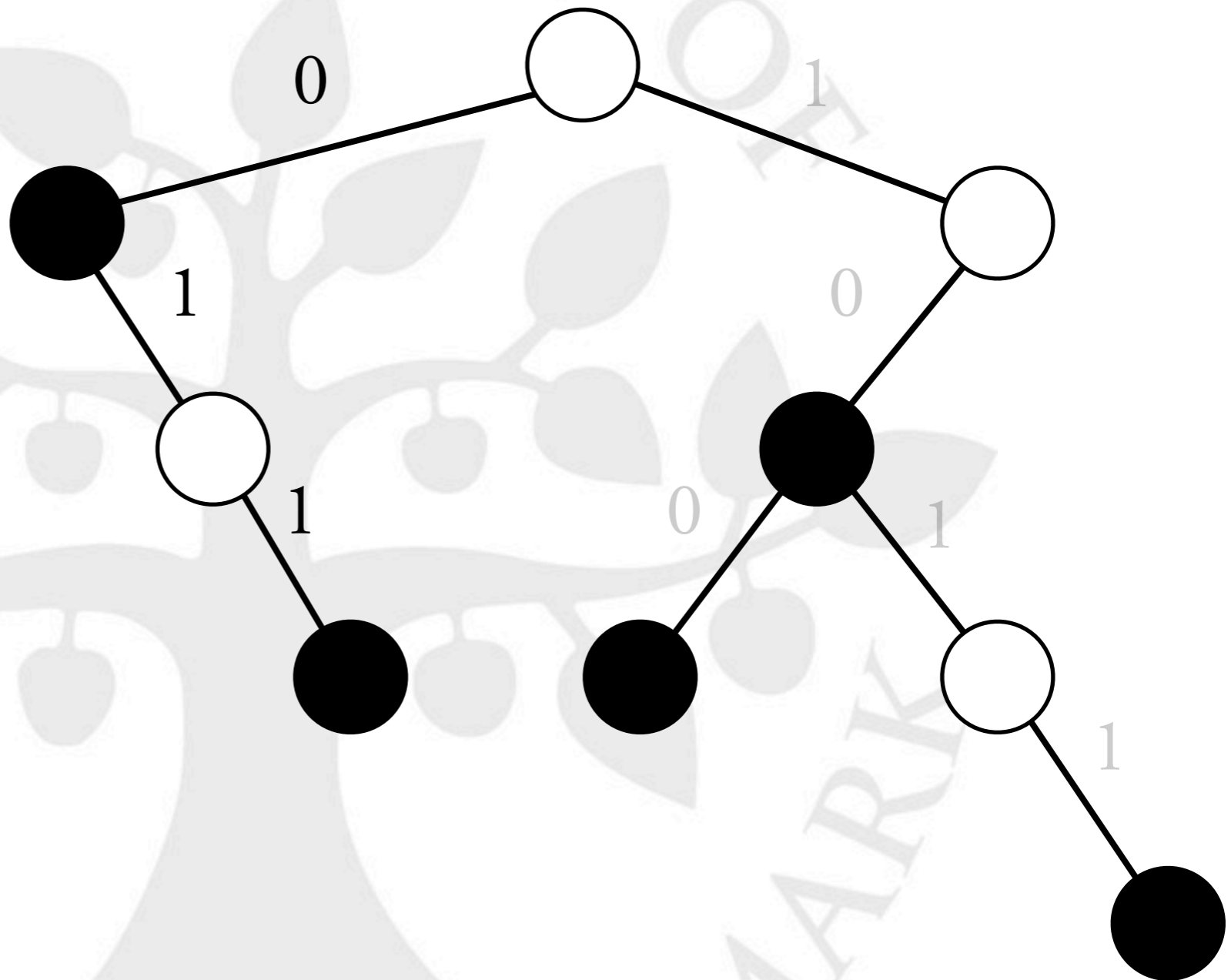
Radix-træer over binære tal

- Tal i træet?
 - 0



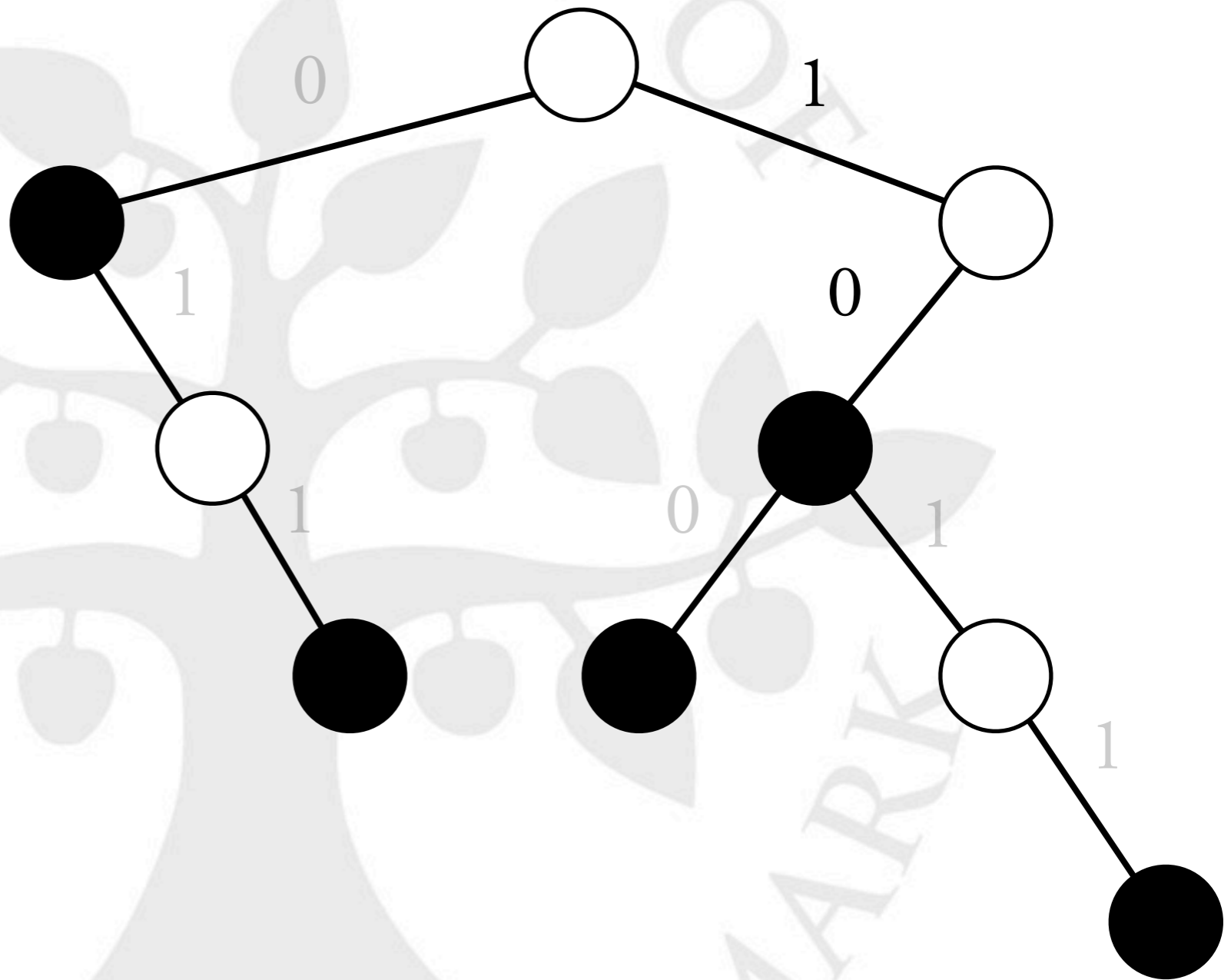
Radix-træer over binære tal

- Tal i træet?
 - 0
 - 011



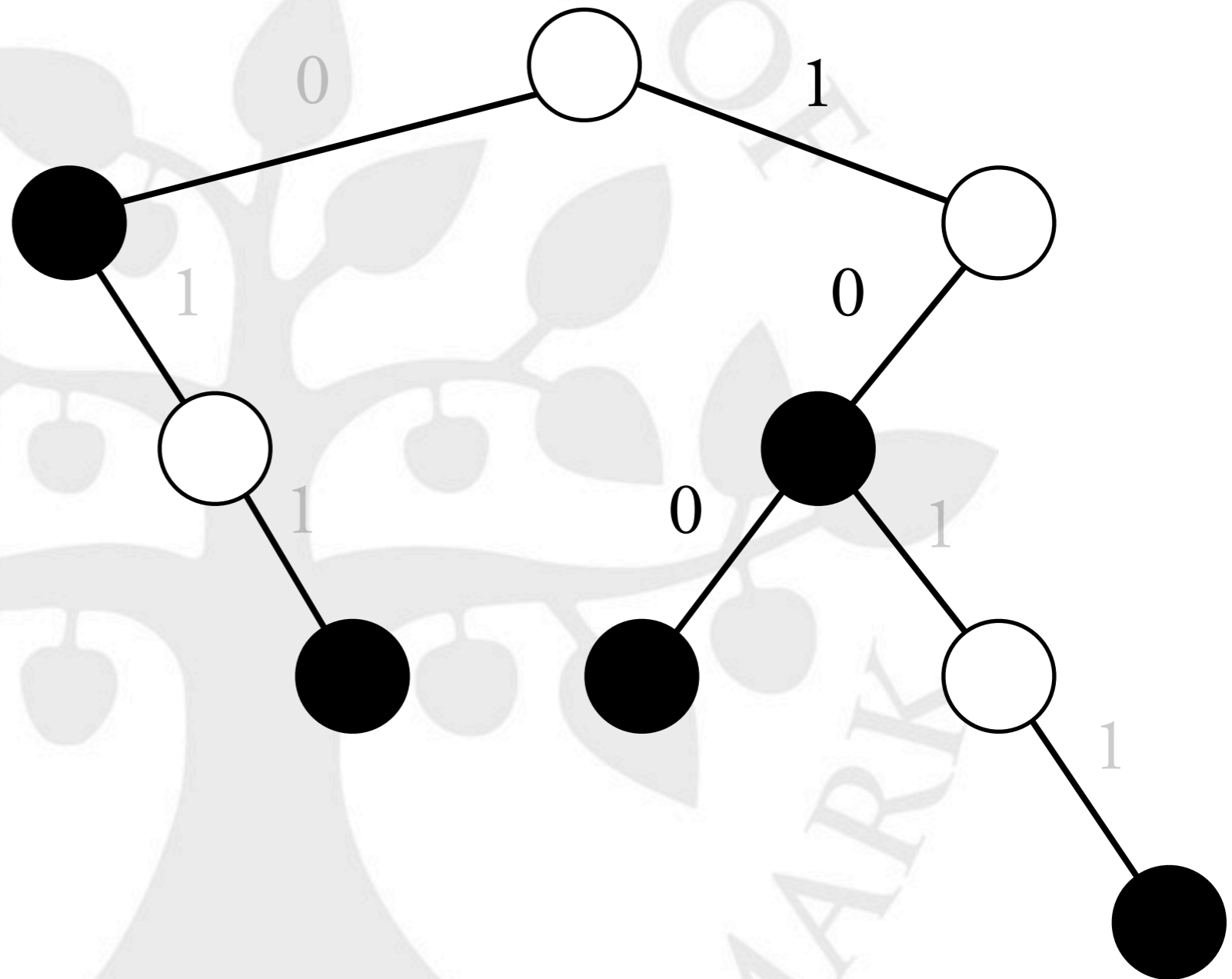
Radix-træer over binære tal

- Tal i træet?
 - 0
 - 011
 - 10



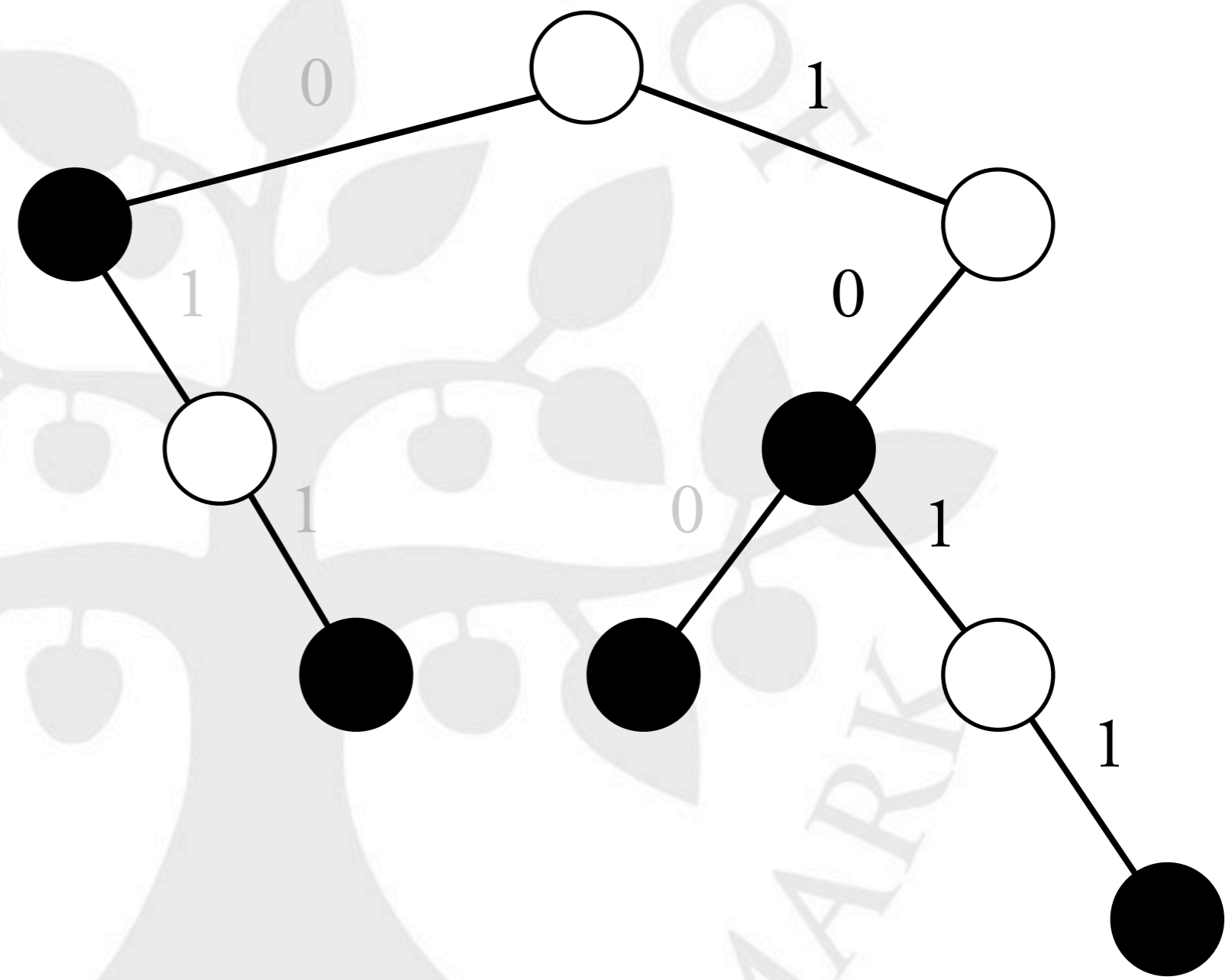
Radix-træer over binære tal

- Tal i træet?
 - 0
 - 011
 - 10
 - 100

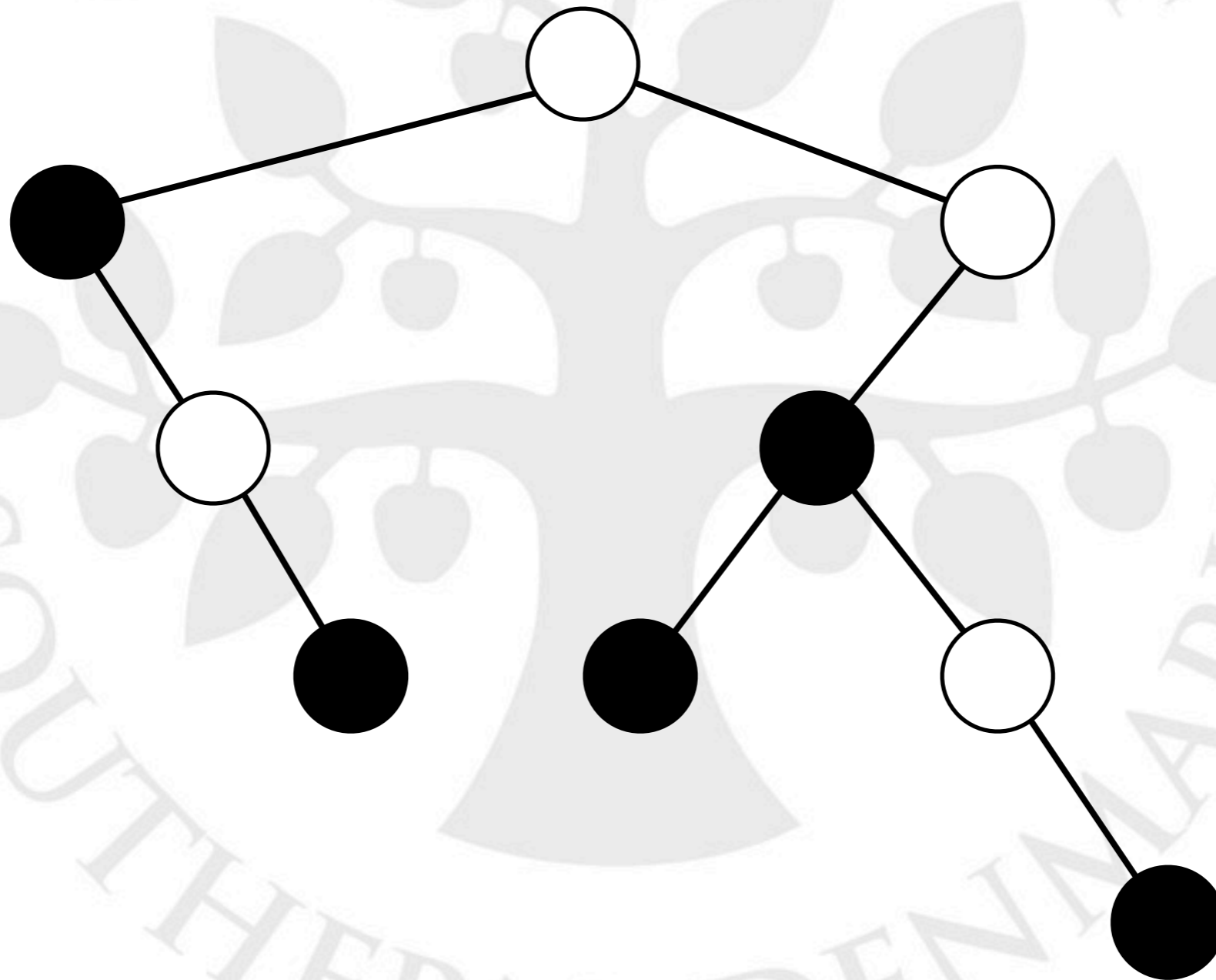


Radix-træer over binære tal

- Tal i træet?
 - 0
 - 011
 - 10
 - 100
 - 1011

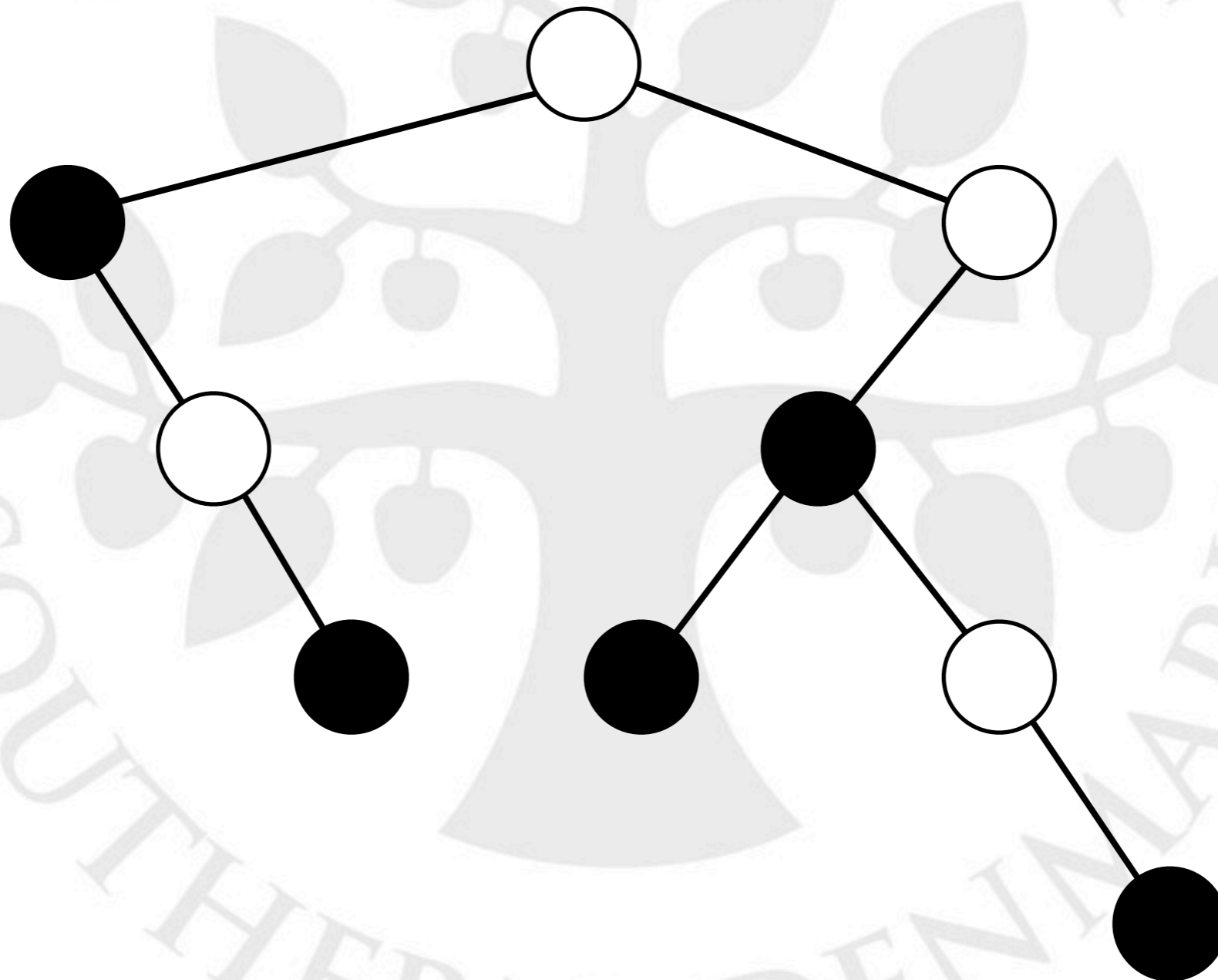


Radix-træer over binære tal



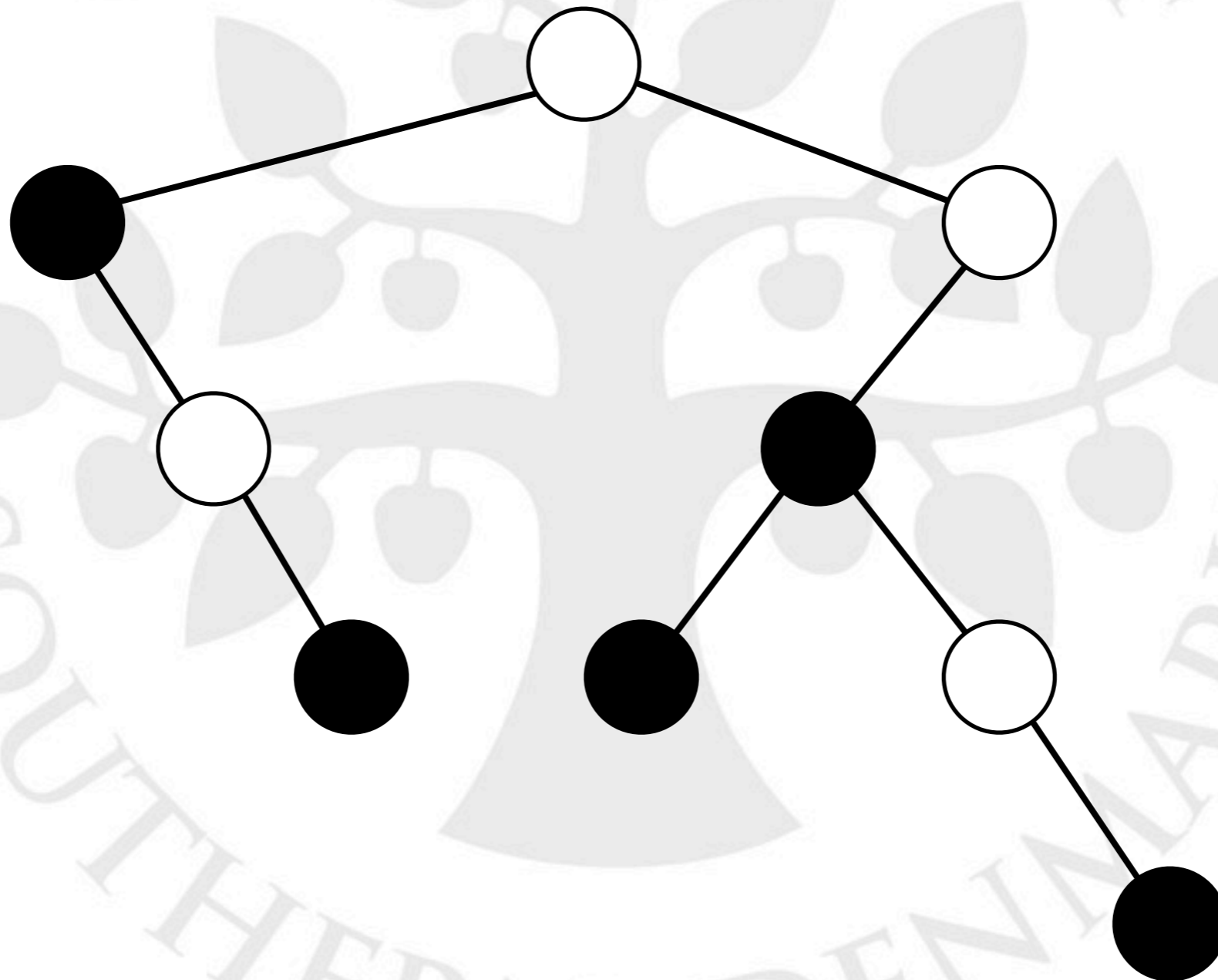
Radix-træer over binære tal

- Indsættelse



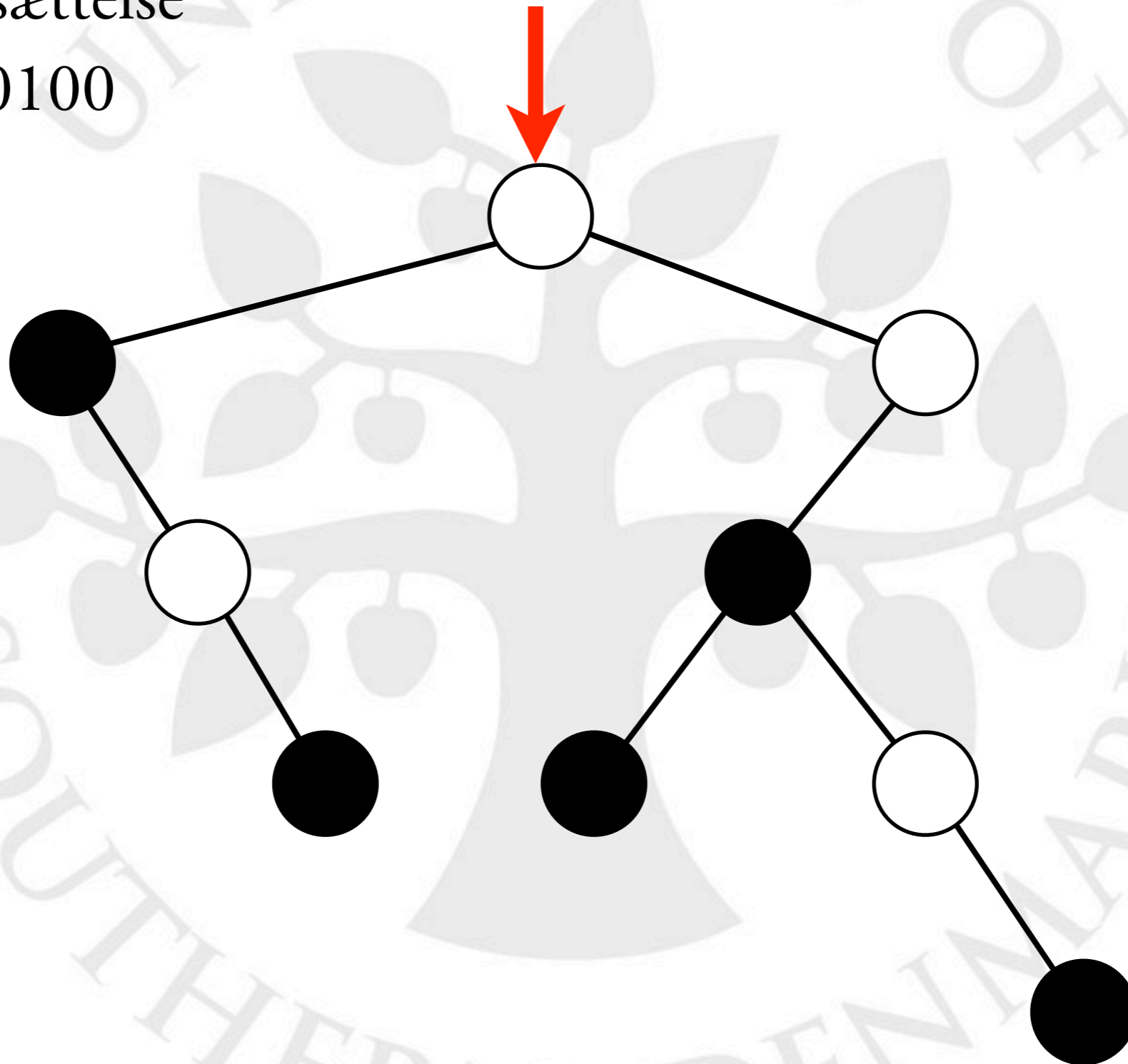
Radix-træer over binære tal

- Indsættelse
 - 0100



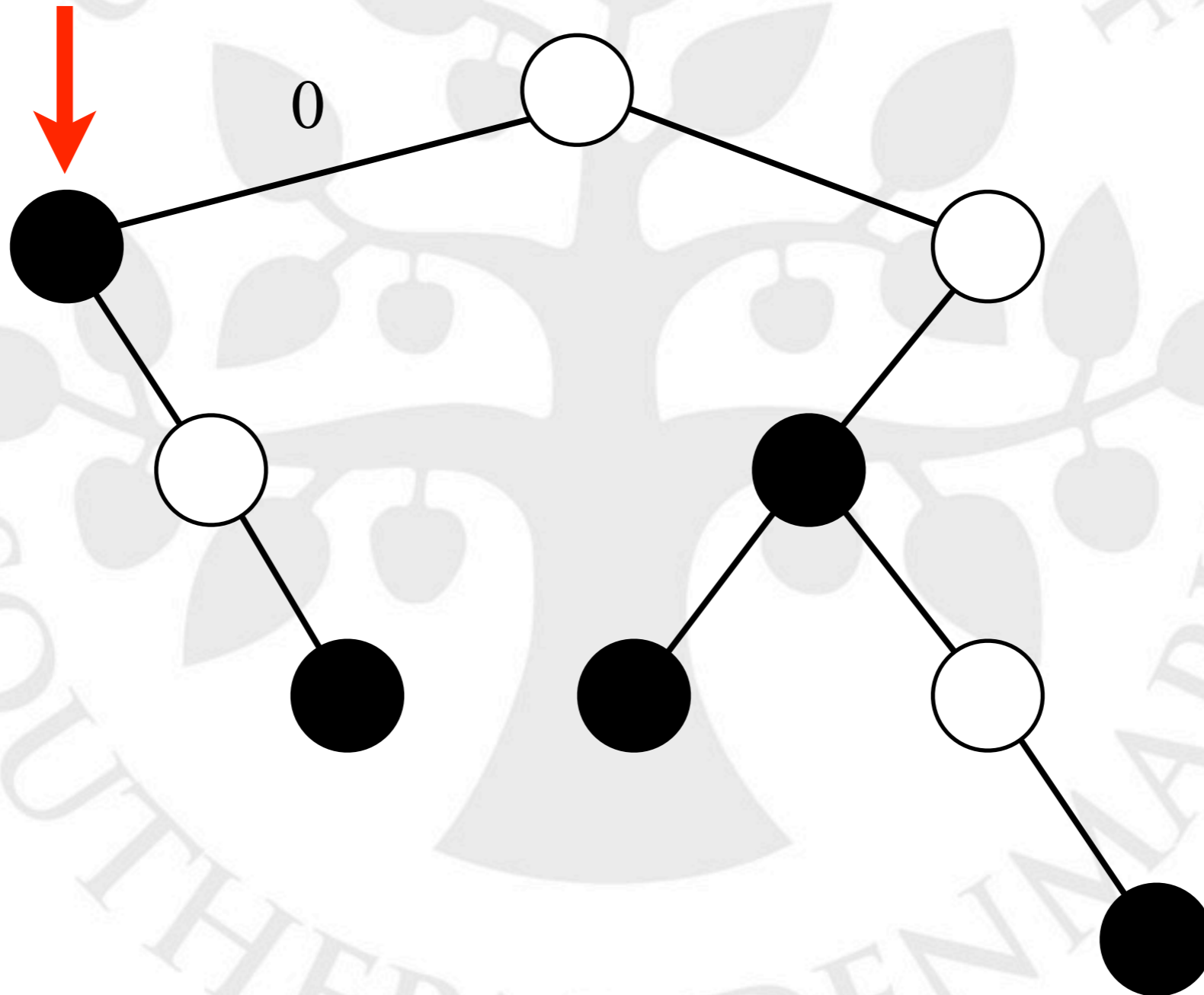
Radix-træer over binære tal

- Indsættelse
 - 0100



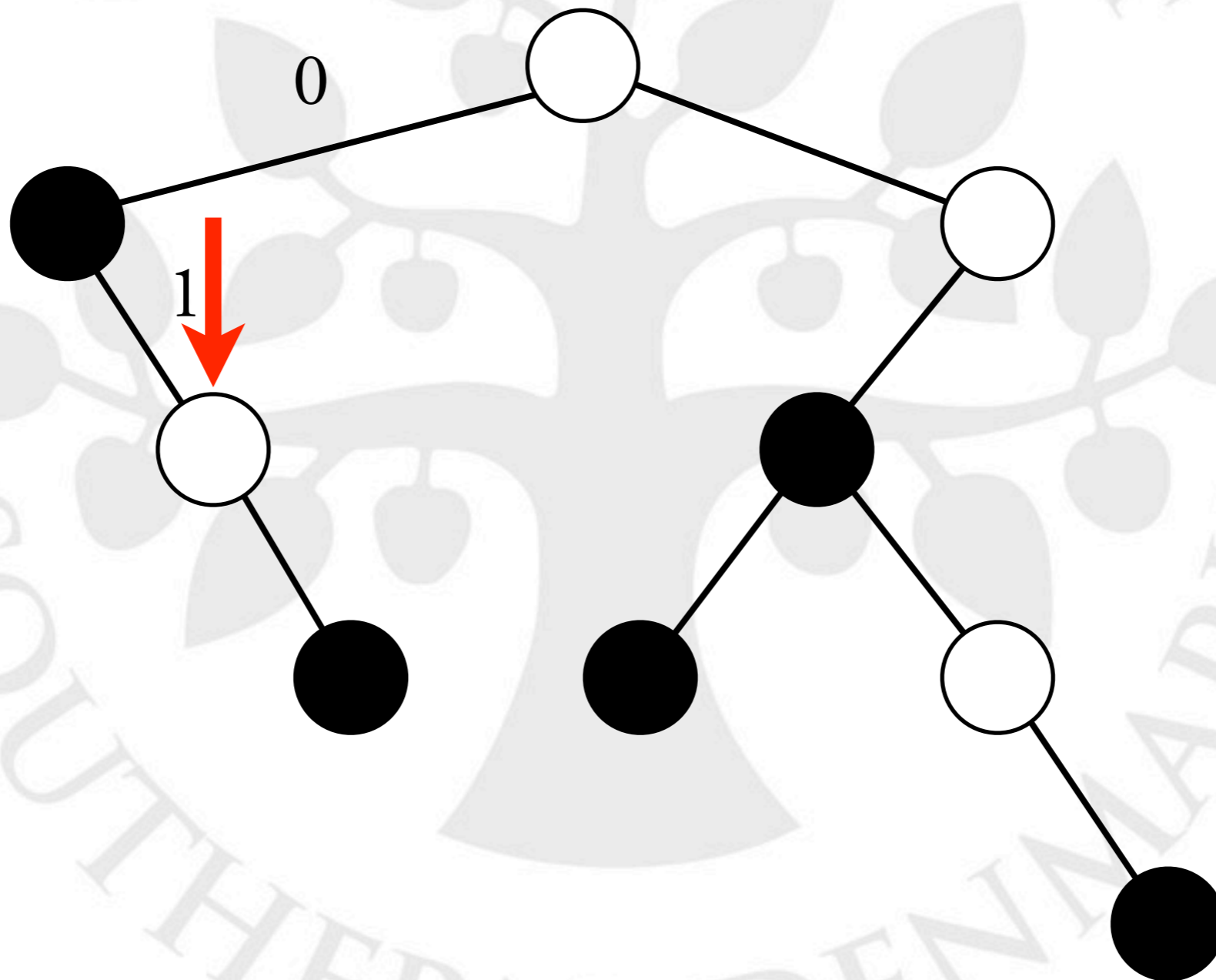
Radix-træer over binære tal

- Indsættelse
 - 0100



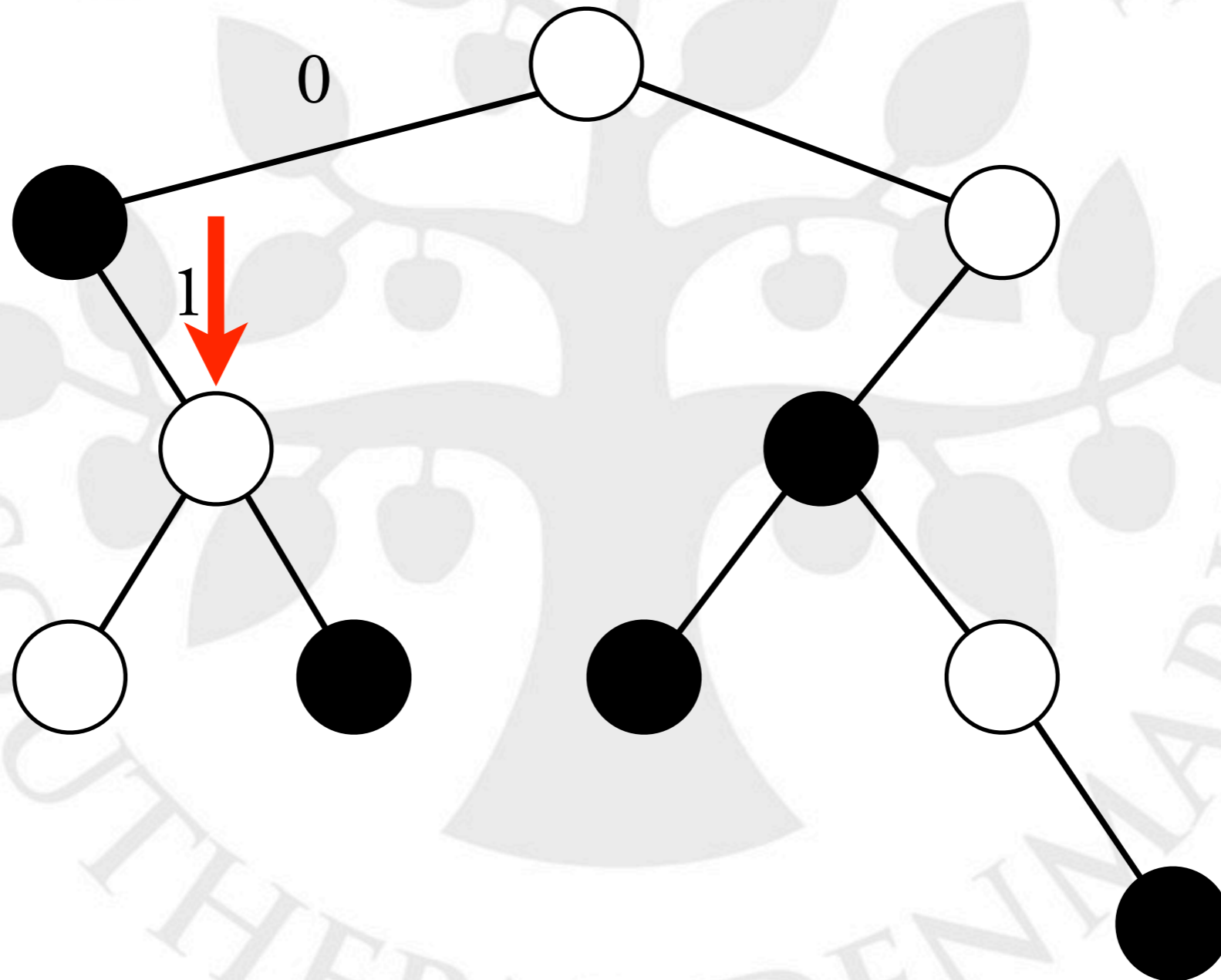
Radix-træer over binære tal

- Indsættelse
 - 0100



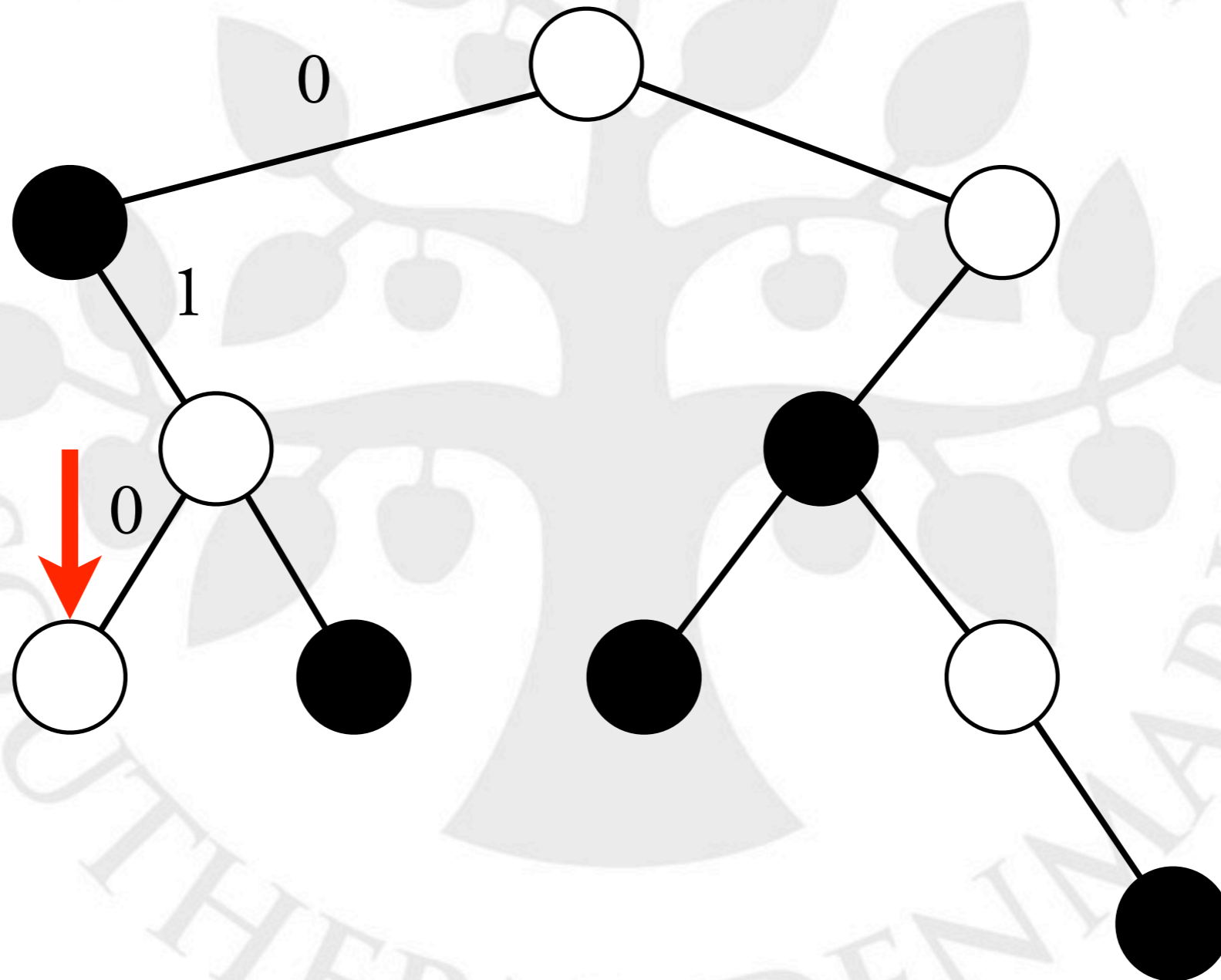
Radix-træer over binære tal

- Indsættelse
 - 0100



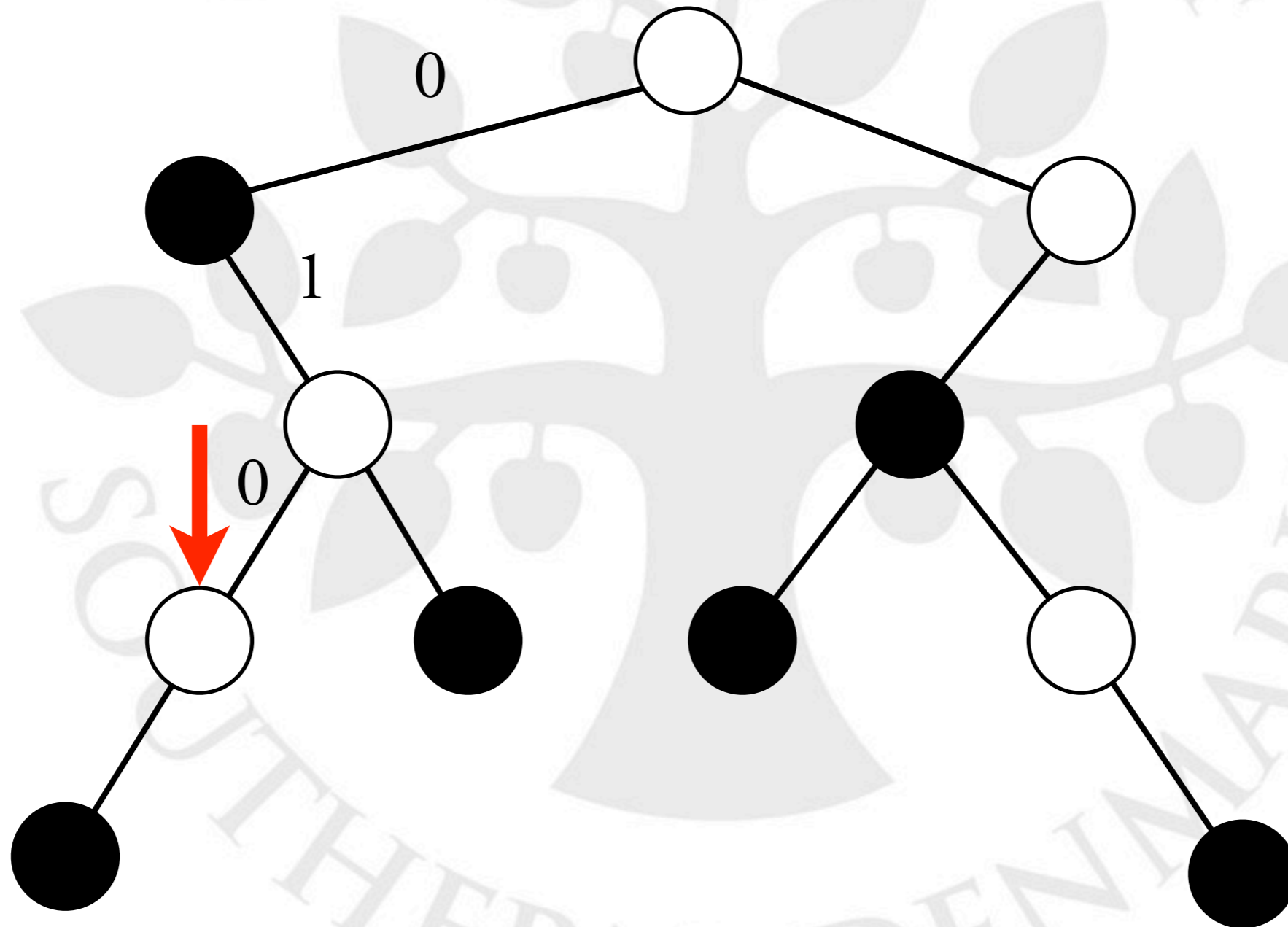
Radix-træer over binære tal

- Indsættelse
 - 0100



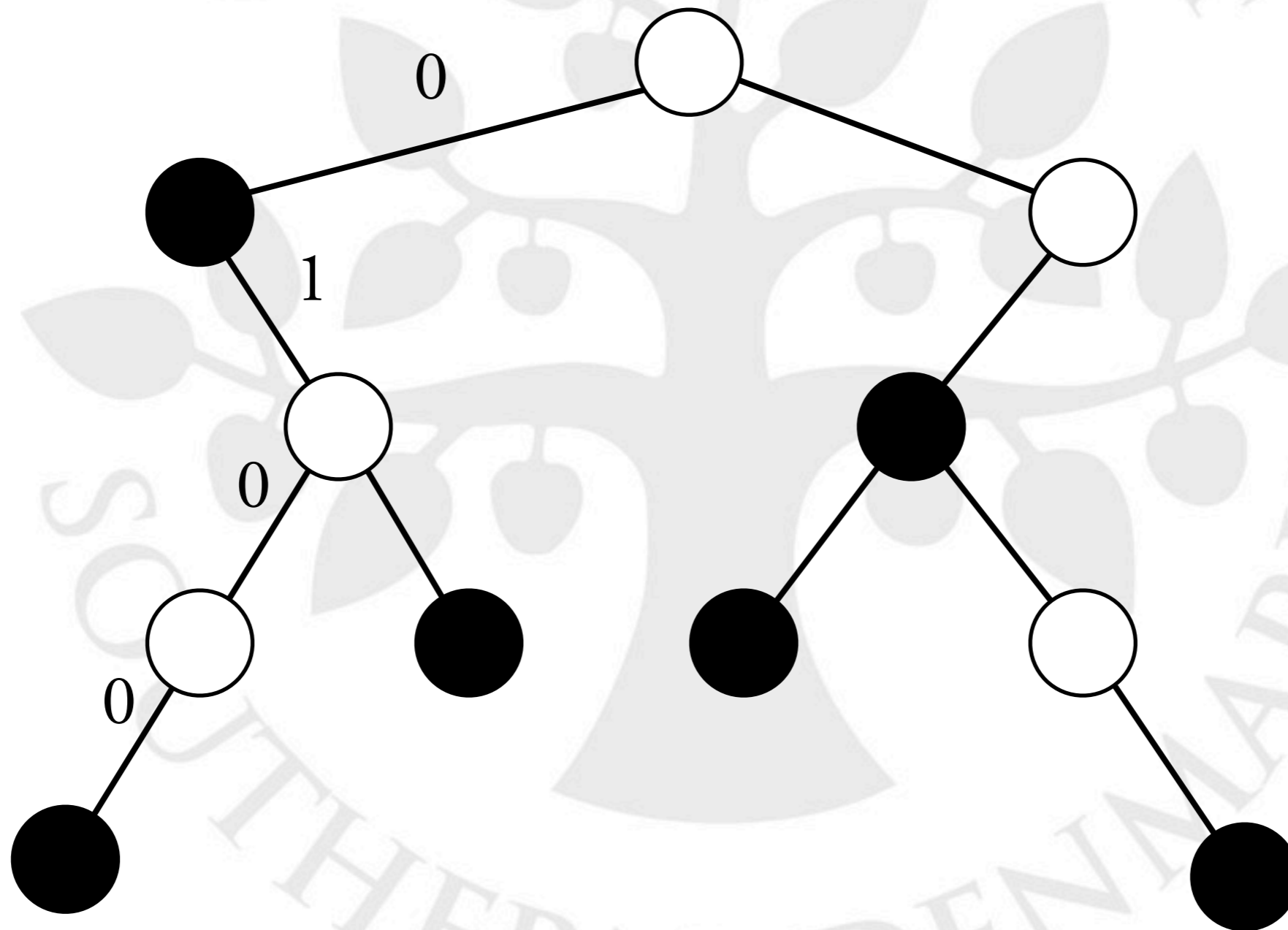
Radix-træer over binære tal

- Indsættelse
 - 0100



Radix-træer over binære tal

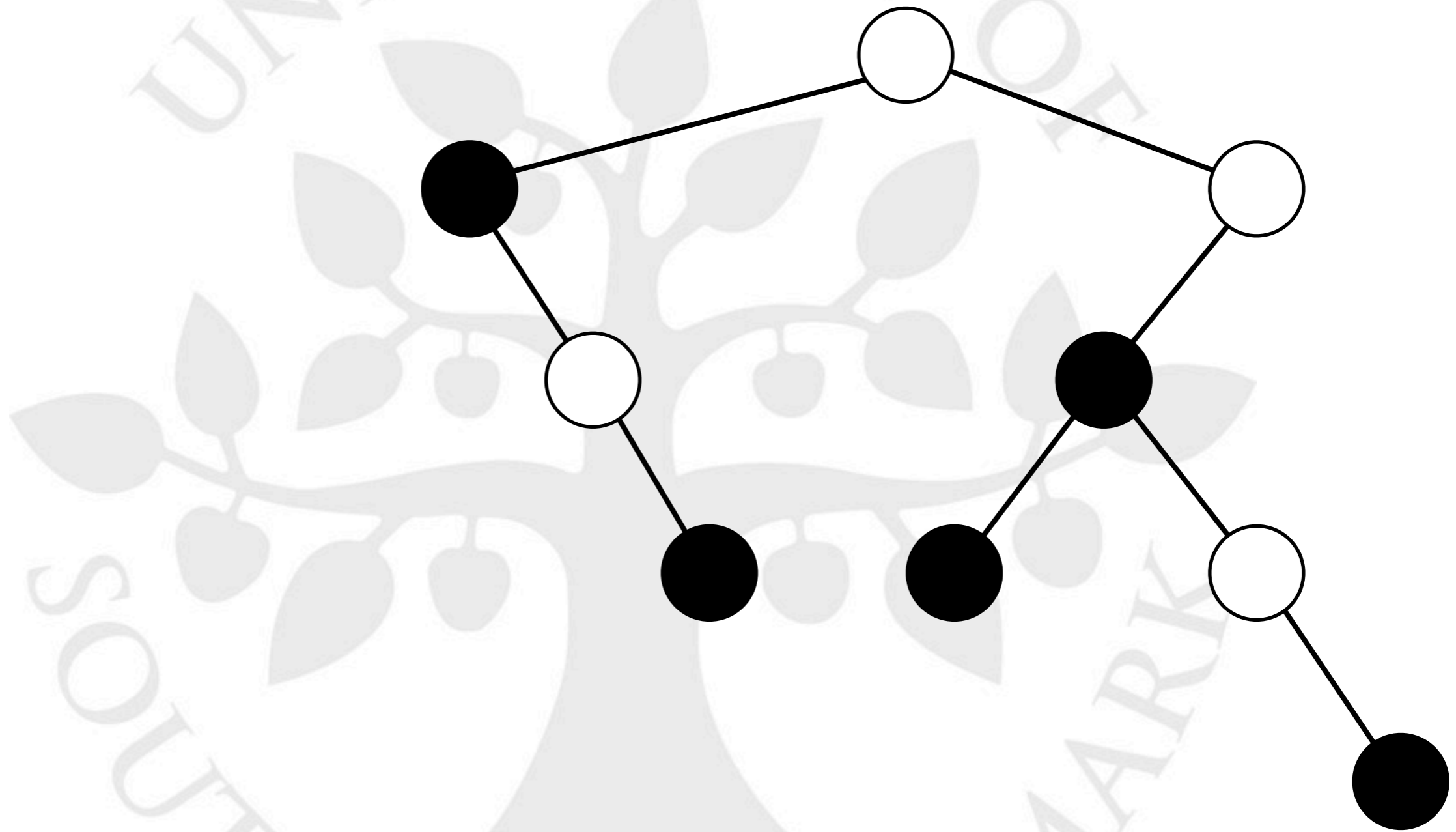
- Indsættelse
 - 0100



Leksikografisk orden/sortering

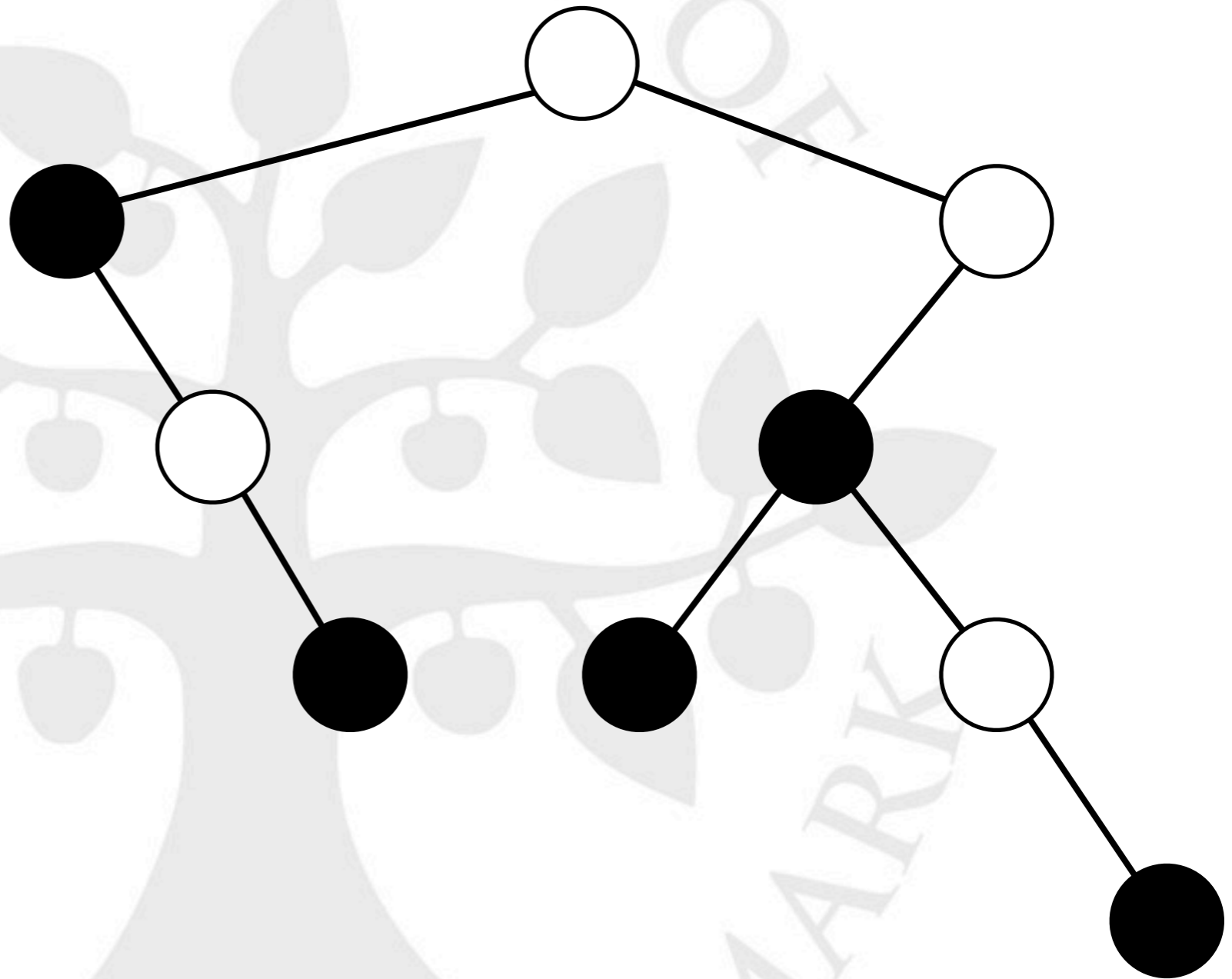
- Ordning der findes i en ordbog
 - $10100 < 10110$
 - De første tre cifre er ens (101)
 - Det næste ciffer i hvert tal er forskelligt
 - Da $0 < 1$ er $10100 < 10110$
 - $10100 < 101000$
 - De første 5 cifre er ens (10100)
 - Der er ikke noget næste ciffer i det første tal, mens der er flere cifre i det andet tal

Leksikografisk og radix-træer



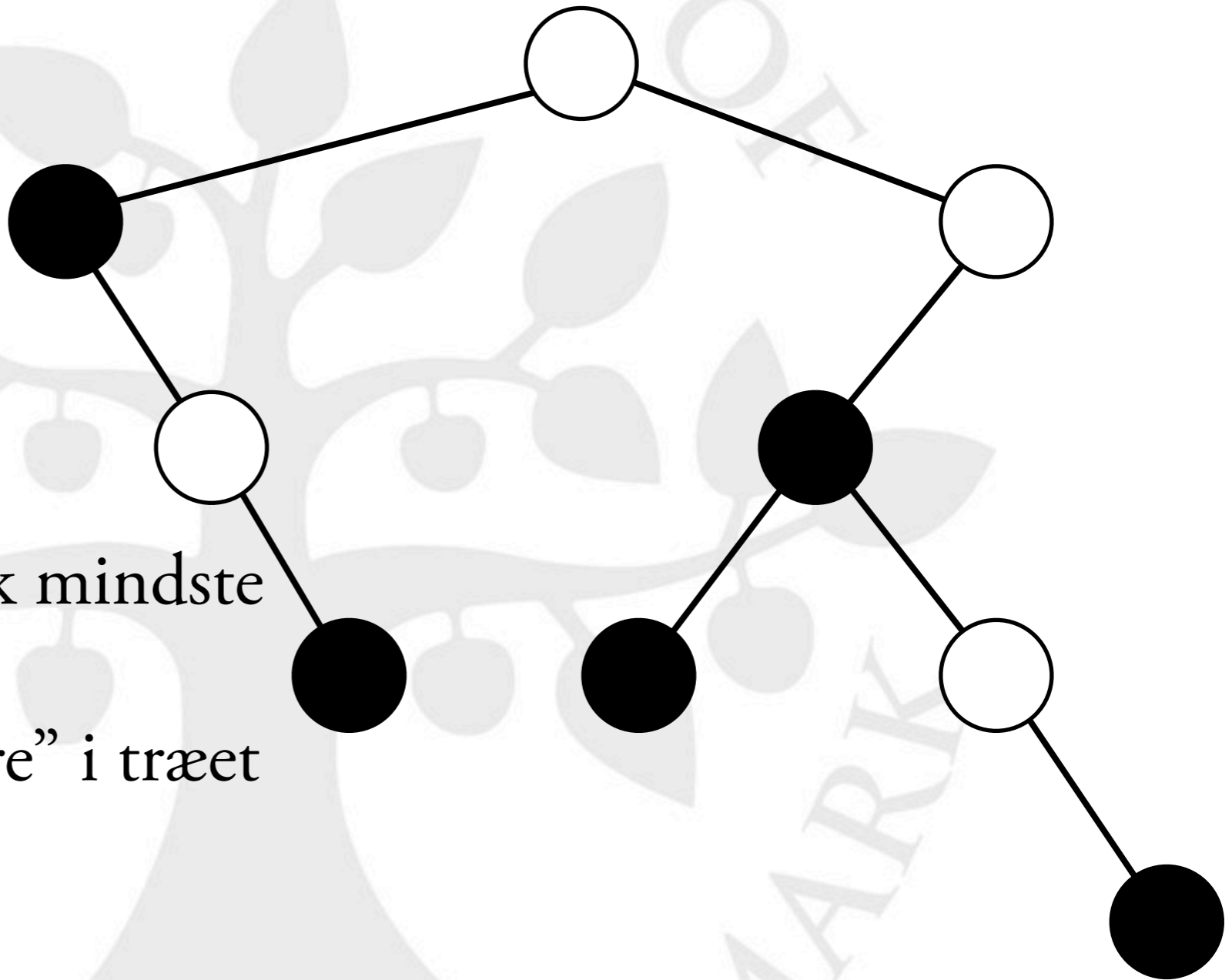
Leksikografisk og radix-træer

- Tal i træet
 - 0
 - 011
 - 10
 - 100
 - 1011



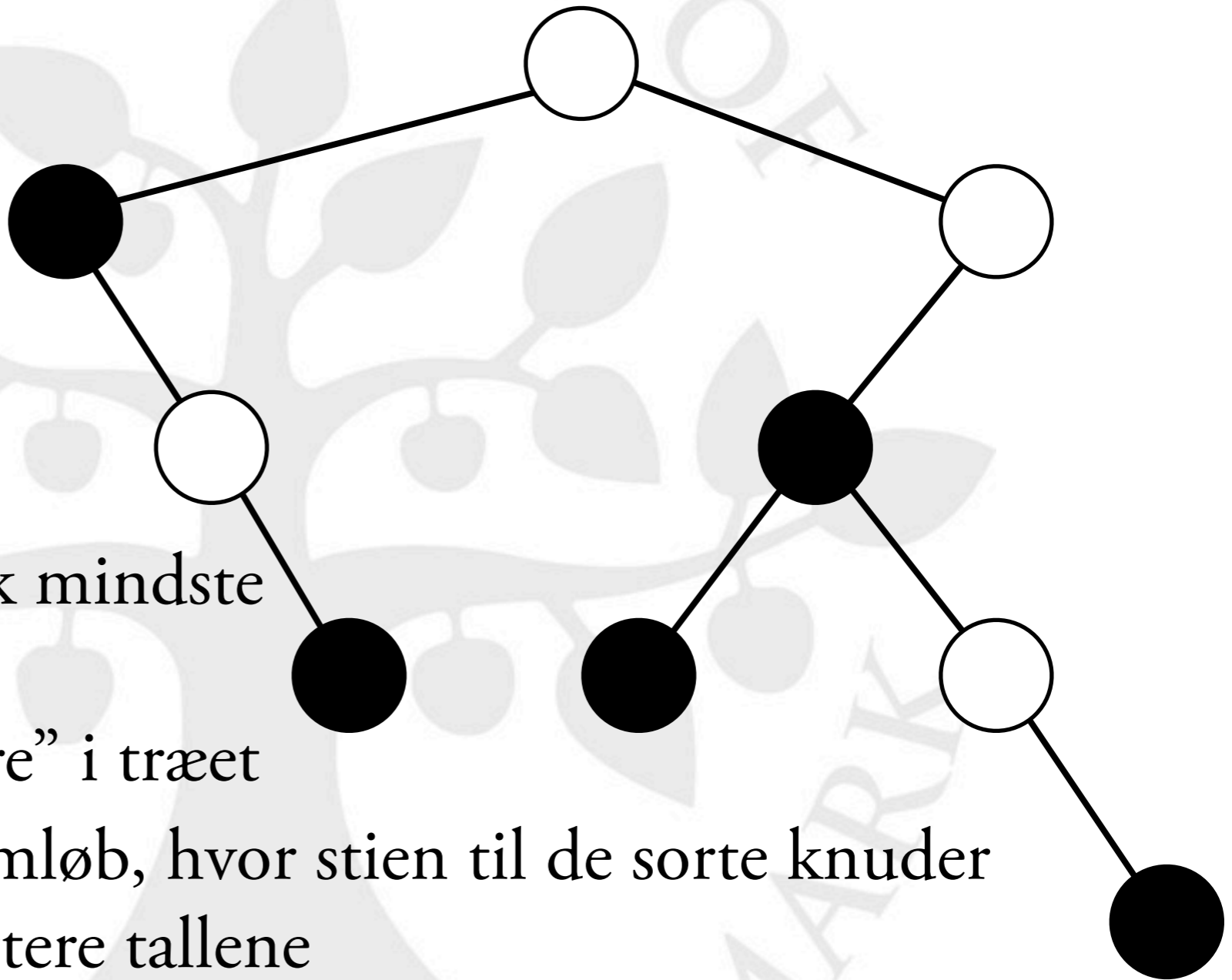
Leksikografisk og radix-træer

- Tal i træet
 - 0
 - 011
 - 10
 - 100
 - 1011
- De leksikografisk mindste tal findes “længst til venstre” i træet



Leksikografisk og radix-træer

- Tal i træet
 - 0
 - 011
 - 10
 - 100
 - 1011
- De leksikografisk mindste tal findes “længst til venstre” i træet
- Pre-order gennemløb, hvor stien til de sorte knuder udskrives, vil sortere tallene



Programmet



Programmet

- To klasser



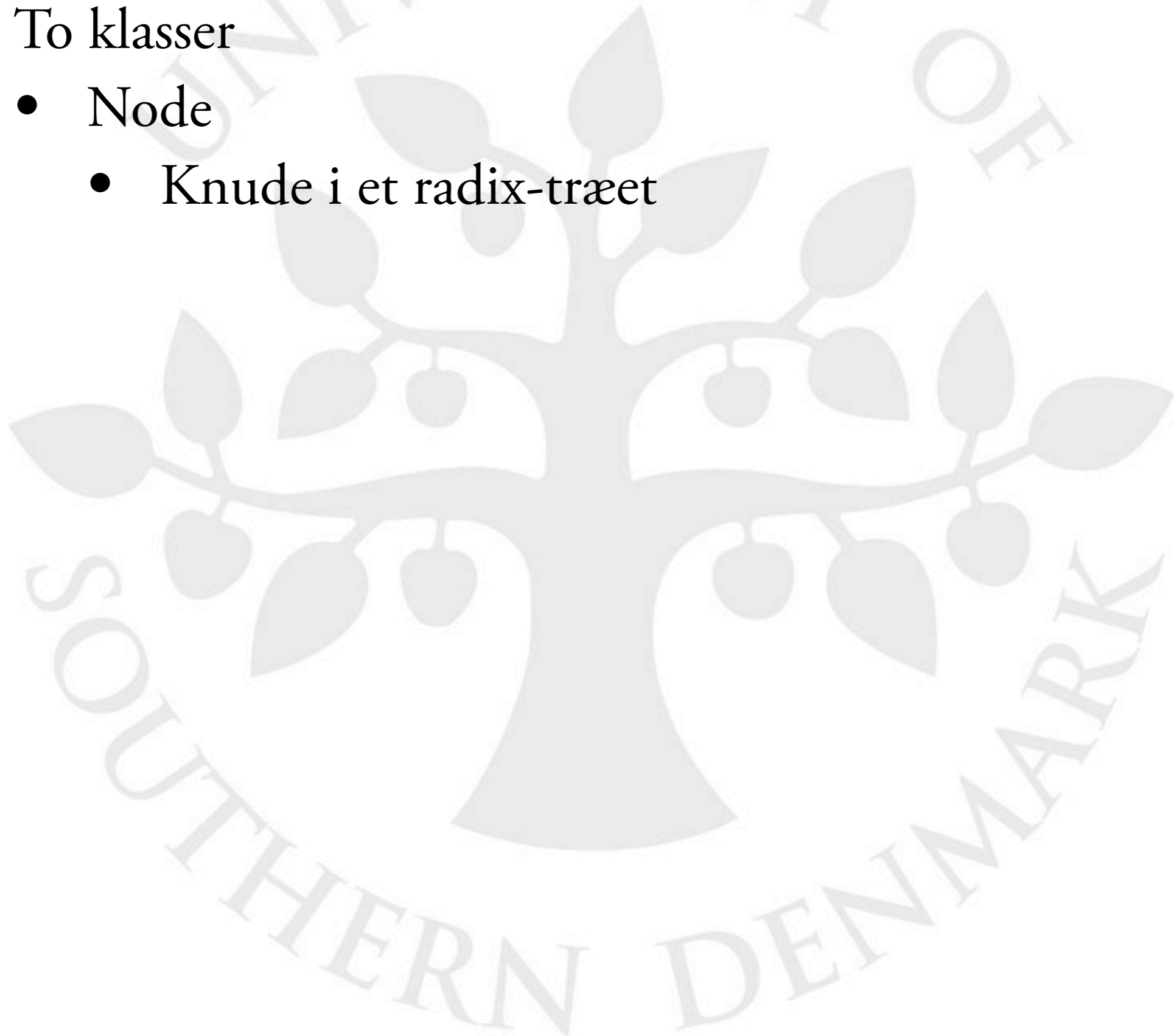
Programmet

- To klasser
 - Node



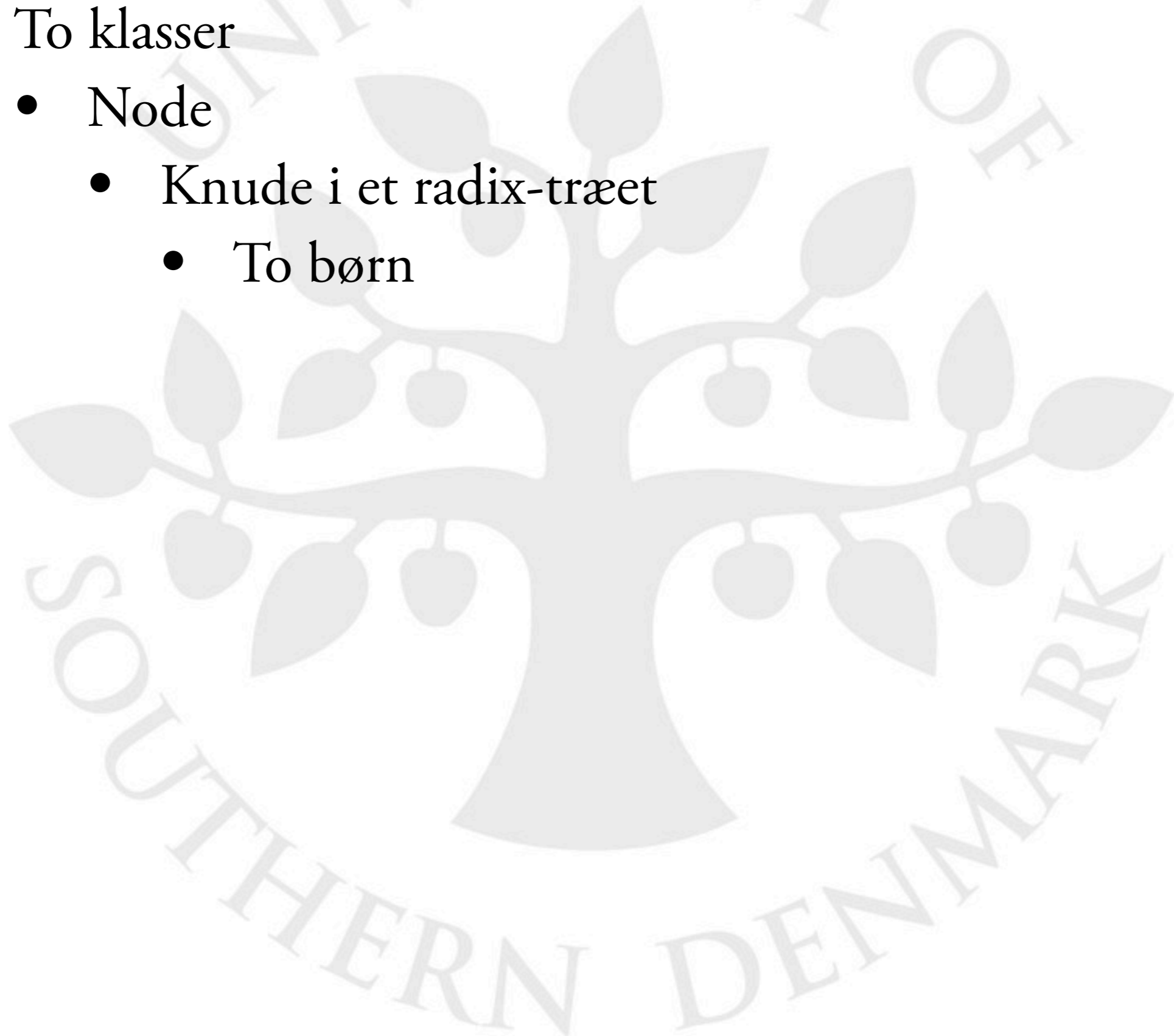
Programmet

- To klasser
 - Node
 - Knude i et radix-træet



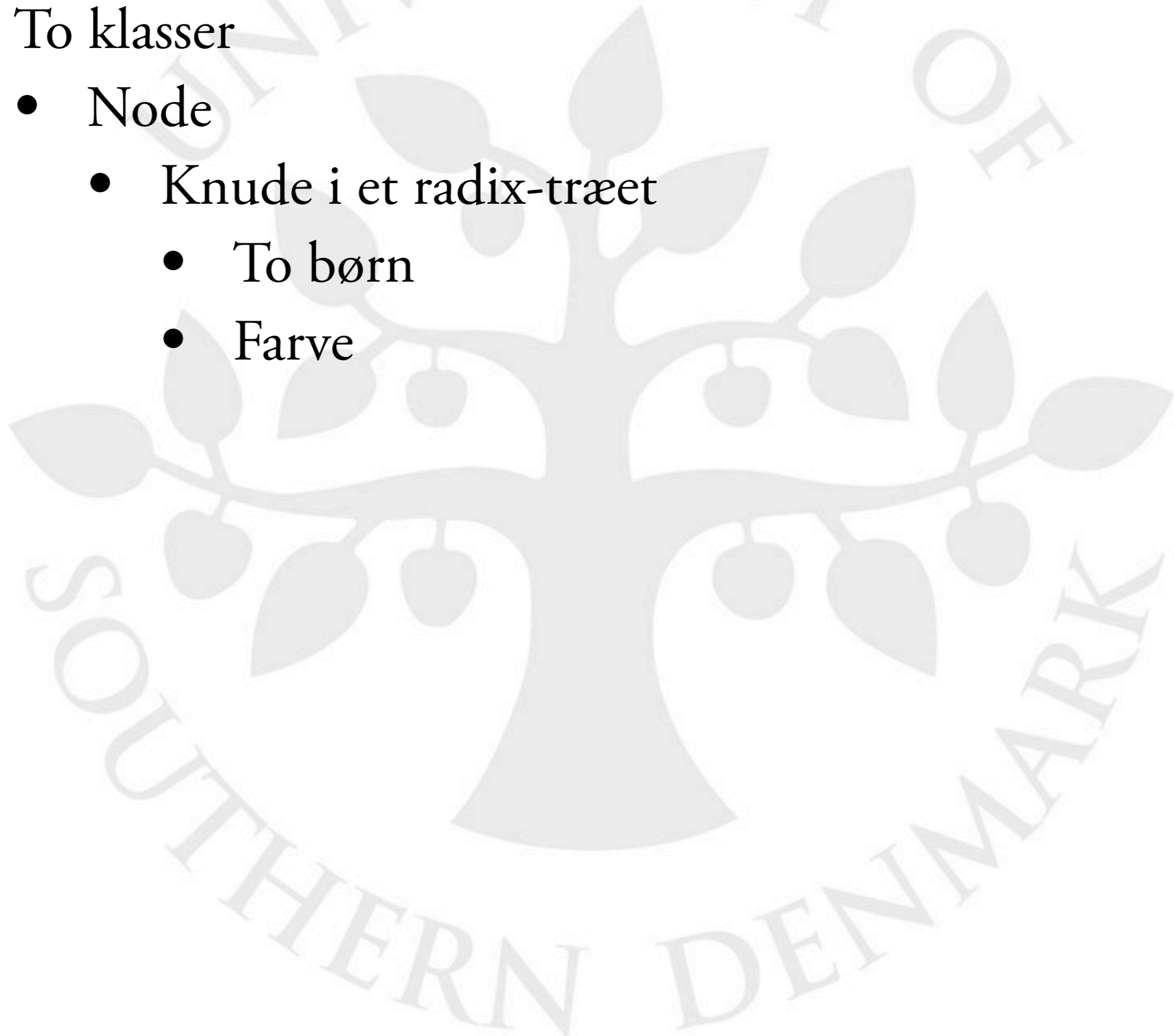
Programmet

- To klasser
 - Node
 - Knude i et radix-træet
 - To børn



Programmet

- To klasser
 - Node
 - Knude i et radix-træet
 - To børn
 - Farve



Programmet

- To klasser
 - Node
 - Knude i et radix-træet
 - To børn
 - Farve
 - RadixSorter



Programmet

- To klasser
 - Node
 - Knude i et radix-træet
 - To børn
 - Farve
 - RadixSorter
 - Læser tallene fra input-filen



Programmet

- To klasser
 - Node
 - Knude i et radix-træet
 - To børn
 - Farve
 - RadixSorter
 - Læser tallene fra input-filen
 - Bygger radix-træet



Programmet

- To klasser
 - Node
 - Knude i et radix-træet
 - To børn
 - Farve
 - RadixSorter
 - Læser tallene fra input-filen
 - Bygger radix-træet
 - Udskriver tallene i leksikografisk orden vha. et pre-order gennemløbe af radix-træet

Programmet

- To klasser
 - Node
 - Knude i et radix-træet
 - To børn
 - Farve
 - RadixSorter
 - Læser tallene fra input-filen
 - Bygger radix-træet
 - Udskriver tallene i leksikografisk orden vha. et pre-order gennemløbe af radix-træet
 - (Anvend gerne egne rekursive metoder på Node)

Rapporten

- Design
 - Hvilke programdele skal der være?
 - Hvilke datastrukturer skal der være?
 - Hvordan skal (hoved)programmet struktureres?
Initialisering, indlæsningsløkke, beregning og udskrift
 - Er der nogen interessante algoritmer?



Rapporten

- Implementation
 - Hvordan har du konkret (med kode) udført designet?
 - Hvordan indlæser du inddata?
 - Kan programmet gå ned? Kan det undgås?
 - Hvordan foretages de konkrete beregninger?
 - Hvordan genereres output?
 - HUSK en komplet udskrift af koden!

Rapporten

- Afprøvning
 - Korrekte data: Typiske data
 - Giver programmet korrekt output?
 - Grænsetilfælde og specialtilfælde såsom intet eller "svært" input
 - Sørg for at lave afprøvninger der kommer rundt i alle dele af koden
 - Problematisk/ukorrekt data: Data der ikke overholder specifikationen
 - Går programmet ned og på hvilken måde?
 - Gør det noget? Og i givet fald, er der noget af gøre ved det?
 - Husk at inkludere samhørende ind- og uddata!

Rapporten

- Konklusion
 - Hvordan forløb testen?
 - Virker programmet korrekt?
 - Hvilke mangler har programmet?
 - Hvad kunne gøres bedre/anderledes?

